



Sistemas Informáticos

Curso 2002-03

Creación de un entorno de clase virtual en Internet

Beatriz Vera Mateos
Esther Valverde Escorial
Jorge Juan Llorente Marcos

Dirigido por:
Prof. Baltasar Fernández Manjón
Dpto. de Sistemas Informáticos y Programación

Facultad de Informática
Universidad Complutense de Madrid

**CREACIÓN DE UN ENTORNO DE CLASE VIRTUAL EN INTERNET****INDICE**

1. RESÚMEN: CREACIÓN DE UN ENTORNO DE CLASE VIRTUAL EN INTERNET.....	5
2. PROJECT'S SUMMARY	6
3. MANUAL DE INSTALACIÓN (APLICACIONES Y LIBRERÍAS)	7
3.1. JDK 1.3.1.02	7
3.2. APACHE 1.3.26.....	8
3.3. TOMCAT 4.0.4.....	9
3.4. TOMCAT COMO SERVICIO.....	11
3.5. COMUNICACIÓN APACHE – TOMCAT	12
3.6. LIBRERÍAS XALAN Y XERCES (JAXP 1.2.01).....	12
3.7. PAQUETE COM.OREILLY.SERVLET	13
3.8. JAVAMAIL 1.2.....	13
3.9. FICHERO DE CONFIGURACIÓN 'CONF.TXT' Y 'FTP.TXT'	13
3.10. CLASEVIRTUAL	14
3.11. POSIBLES PROBLEMAS Y SOLUCIONES	14
3.11.1. Orden de arranque	14
3.11.2. Actualizaciones software	14
3.11.3. Configuración de Apache	15
3.11.4. Logs	15
3.11.5. Classpath.....	15
3.12. DIRECCIONES DE INTERÉS	15
4. INTERFAZ GRÁFICA DE USUARIO.....	16
4.1. INTRODUCCIÓN	16
4.2. MODIFICACIONES.....	16
5. CURSO IMS.....	19
6. ACTUALIZACIÓN AL ESTÁNDAR IMS LEARNING RESOURCE METADATA V1.2.2.....	20
7. METADATOS EN FICHERO EXTERNO	21
7.1. ESTRUCTURA DE META DATOS EN FICHERO EXTERNO	21
7.1.1. Meta datos en el sistema anterior.....	21
7.1.2. Meta datos en el sistema actual.....	22
7.2. IMPLEMENTACIÓN DE META DATOS EN FICHERO EXTERNO	22
7.2.1. Fichero de meta-datos	22
7.2.2. Ficheros imsmanifest.xml y curso.dtd.....	23
7.2.3. Fichero curso.java.....	23
8. TRATAMIENTO DE LOS PRERREQUISITOS DE UN ÍTEM	26
8.1. PRERREQUISITOS EN EL SISTEMA ORIGINAL.....	26
8.1.1. Ejemplo de prerrequisitos en el sistema original	26
8.2. PRERREQUISITOS EN EL SISTEMA ACTUAL	27
8.2.1. Ejemplo de prerrequisitos en el sistema actual	28
9. SUSTITUCIÓN DE DOM POR XPATH.....	29
10. DIRECTRICES FUTURAS: ACTUALIZACIÓN DE PAQUETE DE CONTENIDO A UNIDAD DE APRENDIZAJE SEGÚN EL ESTÁNDAR IMS LEARNING DESIGN	32
10.1. INTRODUCCIÓN	32



10.2.	DESCRIPCIÓN	32
10.3.	COMPARACIÓN CON EL IMS CONTENT PACKAGING Y OTROS ESTÁNDARES	32
10.4.	USO EN NUESTRA APLICACIÓN.	34
10.5.	CUADRO COMPARATIVO ENTRE EL IMSMANIFES.XML EN UN PAQUETE DE CONTENIDO Y EL IMSMANIFEST.XML EN UNA UNIDAD DE APRENDIZAJE	34
10.6.	RELACIÓN DEL ESTÁNDAR CON LOS META-DATOS.....	36
10.6.1.	<i>Estructura de los meta datos</i>	37
11.	ARQUITECTURA Y ENTORNO TECNOLÓGICO	38
11.1.	ARQUITECTURA FÍSICA DEL SISTEMA	40
11.2.	ARQUITECTURA LÓGICA DEL SISTEMA	41
11.3.	PLATAFORMA.....	43
11.3.1.	<i>Entorno de explotación</i>	43
11.3.2.	<i>Entorno de desarrollo</i>	44
11.4.	DISEÑO DE LA BASE DE DATOS	45
11.4.1.	<i>Estructura</i>	45
11.4.2.	<i>ORGANIZACIÓN DEL ARBOL DE DIRECTORIOS</i>	47
12.	APÉNDICE A: COMPARATIVA DE INTERFACES.....	50
13.	APÉNDICE B: IMPLEMENTACIÓN.....	54
13.1.	CAMBIOS EN EL FICHERO PAGINA2.DTD.....	54
13.2.	CAMBIOS EL LOS FICHEROS MOSTRAR_PAGINA_*.XSL	55
14.	APÉNDICE C: CURSO IMS	57
14.1.	INTRODUCCIÓN A LOS ESTÁNDARES	57
14.1.1.	<i>Términos imprescindibles</i>	58
14.1.2.	<i>Grupos y Organizaciones</i>	61
14.2.	¿ QUÉ ES IMS GLOBAL LEARNING CONSORTIUM ?	62
14.3.	PROCESO DE DESARROLLO DE LAS ESPECIFICACIONES	63
14.3.1.	<i>Recomendaciones para una buena especificación</i>	63
14.3.2.	<i>Conceptos de IMS</i>	64
14.3.3.	<i>Estándares de IMS</i>	64
14.4.	VER EXAMEN EN INTERNET	65
14.5.	IMS CONTENT PACKAGING SPECIFICATION.....	65
14.5.1.	<i>IMS Paquete de Contenido (Content Package)</i>	65
14.5.2.	<i>Modelo Conceptual</i>	66
14.5.3.	<i>Ver evaluación en Internet</i>	70
14.6.	IMS LEARNING RESOURCE META-DATA SPECIFICATION	70
14.6.1.	<i>Introducción</i>	70
14.6.2.	<i>Modelo Conceptual</i>	70
14.6.3.	<i>Ver evaluación en Internet</i>	74
14.7.	IMS ENTERPRISE SPECIFICATION.....	74
14.7.1.	<i>Introducción</i>	74
14.7.2.	<i>Casos de uso</i>	74
14.7.3.	<i>Arquitectura Básica</i>	75
14.7.4.	<i>Modelo de Información Básico</i>	76
14.7.5.	<i>Modelo de Envío de Mensajes</i>	77
14.7.6.	<i>Descripción Conceptual de los Objetos de Datos</i>	78
14.7.7.	<i>Estructura Tabular del Paquete Enterprise</i>	79
14.7.8.	<i>Conformidad</i>	80
14.7.9.	<i>Ver evaluación en Internet</i>	81
14.8.	IMS QUESTION & TEST INTEROPERABILITY SPECIFICATION.....	81
14.8.1.	<i>Introducción</i>	81
14.8.2.	<i>Casos de Uso</i>	82
14.8.3.	<i>Preguntas, Ítems y Respuestas</i>	86



14.8.4.	Modelo de Información Básico.....	91
14.8.5.	Descripción Conceptual de los Objetos de Datos	96
14.8.6.	Ver evaluación en Internet.....	98
14.9.	IMS LEARNING DESIGN SPECIFICATION	98
14.9.1.	Introducción.....	98
14.9.2.	Modelo Conceptual.....	101
14.9.3.	Vocabulario Conceptual del Diseño de Aprendizaje (Learning Design).....	107
14.9.4.	Modelo de Información.....	110
14.9.5.	Modelo de Comportamiento	123
14.9.6.	Modelo de Comportamiento (y II)	131
14.9.7.	Extensibilidad	135
14.9.8.	Comparación del "IMS Content Packaging" con otros estándares IMS	136
14.9.9.	Relación del estándar con los Meta Datos	138
14.9.10.	Ver evaluación en Internet.....	140
14.10.	IMS LEARNER INFORMATION PACKAGE SPECIFICATION	140
14.10.1.	Introducción.....	140
14.10.2.	Sistemas de Información del Estudiante	140
14.10.3.	Casos de Uso	143
14.10.4.	Modelo de Información Básico.....	144
14.10.5.	Empaquetamiento de la información del estudiante	146
14.10.6.	Descripción Conceptual de los Objetos de Datos.....	150
14.10.7.	Vocabularios & Taxonomías IMS LIP.....	153
14.10.8.	Descripciones de los Meta-datos	154
14.10.9.	Conformidad	154
14.10.10.	Ver evaluación en Internet.....	155
14.11.	IMS DIGITAL REPOSITORIES INTEROPERABILITY SPECIFICATION	155
14.11.1.	Introducción.....	155
14.11.2.	Arquitectura Funcional.....	155
14.11.3.	Modelo de Referencia	157
14.11.4.	Modelo General de Mensajería	163
14.11.5.	Visión general de alto nivel de los casos de uso de un almacén de aprendizaje	164
14.11.6.	Casos de Uso	166
14.11.7.	Ver evaluación en Internet.....	166
14.12.	IMS REUSABLE DEFINITION OF COMPETENCY OR EDUCATIONAL OBJECTIVE SPECIFICATION	
	166	
14.12.1.	Introducción.....	166
14.12.2.	Discusión del Modelo de Información	167
14.12.3.	Elementos RDCEO.....	170
14.12.4.	Ver evaluación en Internet.....	171
14.13.	IMS SIMPLE SECUENCING SPECIFICATION	171
14.13.1.	Introducción.....	171
14.13.2.	Componentes de la Secuenciación Simple IMS (IMS Simple Sequencing Components)	
	172	
14.13.3.	Secciones de la Especificación de la Secuenciación Simple (Simple Sequencing Specification Sections)	172
14.13.4.	Suposiciones.....	174
14.13.5.	SM. Modelo de definición de la secuenciación (Sequencing Definition Model)	178
14.13.6.	TM. Modelo de Seguimiento (Tracking Model)	181
14.13.7.	AM. Modelo del Estado de la Actividad (Activity State Model).....	183
14.13.8.	OP. Proceso de Secuenciación Total o Global (Overall Sequencing Process)	185
14.13.9.	NB. Modelo de Comportamiento en la Navegación (Navigation Behaviour Model)...	186
14.13.10.	TB. Modelo de Comportamiento de Finalización (Termination Behaviour Model)	187
14.13.11.	RB. Modelo de Comportamiento de Enrollado (Rollup Behaviour Model)	190
14.13.12.	SR. Modelo de Comportamiento de Selección y Aleatoriedad (Selection and Randomization Behaviour Model)	195
14.13.13.	SB. Modelo de Comportamiento en Serie (Sequencing Behaviour Model).....	197



14.13.14.	DB. Modelo de Comportamiento de Entrega (Delivery Behavior Model).....	208
14.13.15.	UP. Procesos de Utilidad (Utility Processes).....	209
14.13.16.	Ver evaluación en Internet.....	212
14.14.	BIBLIOGRAFIA.....	212
15.	APÉNDICE D: PROCEDIMIENTOS A MODIFICAR SI SE ADOPTA EL ESTÁNDAR IMS LEARNING DESIGN	213
16.	APÉNDICE E: BIBLIOGRAFÍA.....	215
17.	APÉNDICE F: AUTORIZACIÓN	216
18.	APÉNDICE G: LISTA DE PALABRAS CLAVE.....	217



1. RESÚMEN: CREACIÓN DE UN ENTORNO DE CLASE VIRTUAL EN INTERNET.

El proyecto consiste en la creación de una clase virtual en Internet mediante la aplicación de la tecnología XML y usando los estándares educativos IMS.

Se nos entregó una versión anterior del proyecto y nuestra labor se ha centrado en mejorar y retocar dicha aplicación en los siguientes campos:

- Realización de un manual de instalación completo, revisando y mejorando el manual anterior y realizando una recopilación de todos los archivos necesarios para la instalación del proyecto.
- Mejora de la interfaz gráfica, basándonos en la estructura inicial y realizando los cambios oportunos para ofrecer una visión más actualizada.
- Realización de un curso actualizado sobre los diversos estándares IMS, que se ha realizado desde cero ya que el curso que se encontraba en la versión anterior estaba anticuado (los estándares sobre los que hacía una breve introducción han cambiado de versión) por lo que era inservible.
- Se ha llevado a cabo la actualización del proyecto para adaptarlo al nuevo estándar IMS Learning Resource Metada.
- Adaptación del proyecto para su funcionamiento haciendo uso de los Meta-datos en un fichero externo, lo que supuso la traducción y comprensión de los estándares relacionados así como la investigación en Internet de posibles soluciones a este problema, basándonos al final en la encontrada en la página de la organización ADL SCORM, por ser esta una de las más importantes y con experiencia en este campo.
- Como el archivo imsmanifest.xml (manifiesto del curso), se simplificó mucho a realizar los cambios indicados en el punto anterior, se considero interesante terminar de simplificar dicho fichero cambiando el criterio para el tratamiento de los prerrequisitos en los diferentes ítems del curso.
- Mejora de la eficiencia del sistema mediante el uso de XPath, dado que así son más eficientes los recorridos por los ficheros y por tanto, se han actualizado los servlets para conseguir este fin.
- Tras la aparición del nuevo estándar IMS Learning Design, se realizó un estudio minucioso para la adaptación del proyecto a dicho estándar, que se entrega junto al resto de la documentación para un posible uso futuro cuando se establezca y se generalice el uso de dicho estándar a nivel mundial.



2. PROJECT'S SUMMARY

The project consist of making a virtual class on the Internet using the XML technology and the IMS educational specifications.

We started with a previous version of the project and our work has been focused on improve and modify it in the next fields:

- To create a complete installation manual, checking and improving the previous one and making a summary of the necessary files for the project installation.
- We have improve the GUI, having into account the previous one and making changes in order to offer a more actual sight of the application.
- To create a current curse about the differents IMS specifications, that has been maked from zero.
- We have updated the project in order to it fits the IMS Learning Resource Metadata.
- To adapt the project for working with Metadata in a external file. To do this we have translated and understood severals IMS specifications in the same way as we have looked for posible solutions on the Internet. Finally we took the ADL SCORM's solution.
- The imsmanifest.xml file (course's manifest) was simplified thanks to the previous changes, for that reason, we decided to change the way in that the item's requirements are treated for finishing this simplification.
- To improve the system's efficiency by the use of XPath, because in this way, the paths in the files are more efficient.
- After the appearance of the new IMS Learning Design specification, we made a meticulous study for the adaptation of the proyect to this standard. All documentation of this study is delivered with the rest of the project for a possible future use when the standard will be more regular and general all over the world.



3. MANUAL DE INSTALACIÓN (APLICACIONES Y LIBRERÍAS)

Antes de empezar a instalar, ten en cuenta un consejo que te puede ser muy útil para evitar problemas: Siempre que copies un fichero de los que te vamos a ir indicando a lo largo del manual, asegúrate de cambiar las rutas. Si estas instalando en otra unidad distinta de C:\, cámbialo por D:\ o la unidad que estés utilizando. Si en vez de instalar el jdk1.3.1_02, instalas otra versión, asegúrate de cambiarlo en todos los ficheros que sea necesario. Si instalas el Tomcat4 en otro directorio, cámbialo también. **NO TE OLVIDES!!!!**

Para simular la ruta hemos utilizado lo siguiente...

- Para el Tomcat {TOMCAT} → que en nuestro caso se corresponde con C:/Tomcat4
- Para el Apache {APACHE} → que en nuestro caso se corresponde con C:/Archivos de Programa/Apache Group/Apache

3.1. JDK 1.3.1.02

Para instalarlo, ejecutar el fichero *J2SDK_1_3_1_02_WIN.EXE* del CD del proyecto. También se puede conseguir en la página de Java de Sun Microsystems <http://java.sun.com>. Por defecto se instala en el directorio C:\jdk1.3.1_02.

Una vez instalado, hay que añadir al path el directorio de ejecutables del jdk, para ello:

- Windows 9X/ME: debemos añadir la siguiente línea en el fichero autoexec.bat:
set path=%PATH%; C:\jdk1.3.1_02\bin
- Windows NT/2000/XP: modificaremos las variables de entorno (clic con el botón derecho en *MiPC*, pinchamos en *Propiedades*. Seleccionamos la pestaña *Opciones avanzadas* y pulsamos en el botón *Variables de entorno*). Seleccionamos la variable Path y pinchamos en el botón Modificar. Añadimos al final del path: *;C:\jdk1.3.1_02\bin*



3.2. **APACHE 1.3.26**

Para instalarlo, ejecutar el fichero *APACHE_1326_WIN32_X86_SRC.MSI* del CD del proyecto. También se puede conseguir en la página de Apache <http://apache.org>.

El fichero principal de configuración de Apache se llama *httpd.conf* y está ubicado dentro del directorio {APACHE}\conf.

Donde {APACHE} es la ruta donde hemos instalado la aplicación. NOTA: En el fichero *httpd.conf* se puede fijar bajo la directiva *ServerName* la DNS del servidor (o en su defecto la dirección IP del ordenador) y también se puede fijar (bajo la directiva *Port*) el puerto que se está utilizando (por defecto el 80).

A continuación se puede ver un segmento de *httpd.conf* dónde se encuentra los dos valores antes mencionados:

```
.....
#
# ServerName allows you to set a host name which is sent back to clients for
# your server if it's different than the one the program would get (i.e., use
# "www" instead of the host's real name).
#
# Note: You cannot just invent host names and hope they work. The name you
# define here must be a valid DNS name for your host. If you don't understand
# this, ask your network administrator.
```



```
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address (e.g., http://123.45.67.89/)
# anyway, and this will make redirections work in a sensible way.
#
# La dirección IP 127.0.0.1 hace referencia al localhost (ordenador local)
ServerName 127.0.0.1
.....
#
# Port: The port to which the standalone server listens. For
# ports < 1023, you will need httpd to be run as root initially.
#
Port 80
.....
```

Para comprobar que Apache está funcionando correctamente:

1. Ejecuta Inicio->Programas ->Apache HTTP Server->Control Apache Server->Start.
2. Abre una ventana de Internet Explorer e introduce la dirección `http://127.0.0.1`
3. Deberías ver una página de Apache en la que dice que el servidor Apache está funcionando correctamente.
4. Para parar Apache debes ejecutar Inicio->Programas ->Apache HTTP Server->Control Apache Server->Stop.

Los ficheros html que sirve Apache se encuentra bajo el directorio {APACHE}\htdocs

La instalación de Apache como servicio es directa.

3.3. TOMCAT 4.0.4

Para instalarlo, ejecutar el fichero `JAKARTA_TOMCAT_404.EXE` del CD del proyecto. También se puede conseguir en la página de Tomcat <http://jakarta.apache.org>.

Lo único que debes tener en cuenta es que el directorio donde se instala debe de ser un nombre de MS-DOS, es decir, no debe de contener más de 8 caracteres. Un buen nombre y el que usaremos en los ejemplos es el de "tomcat4".

Hay que configurar las variables de entorno de tomcat:

- Windows 9X/ME: Edita el fichero {TOMCAT}\bin\catalina.bat y añade 2 líneas que definan las variables de entorno `JAVA_HOME` y `CATALINA_HOME` de manera que apunten al directorio donde se encuentra el JDK y Tomcat respectivamente.

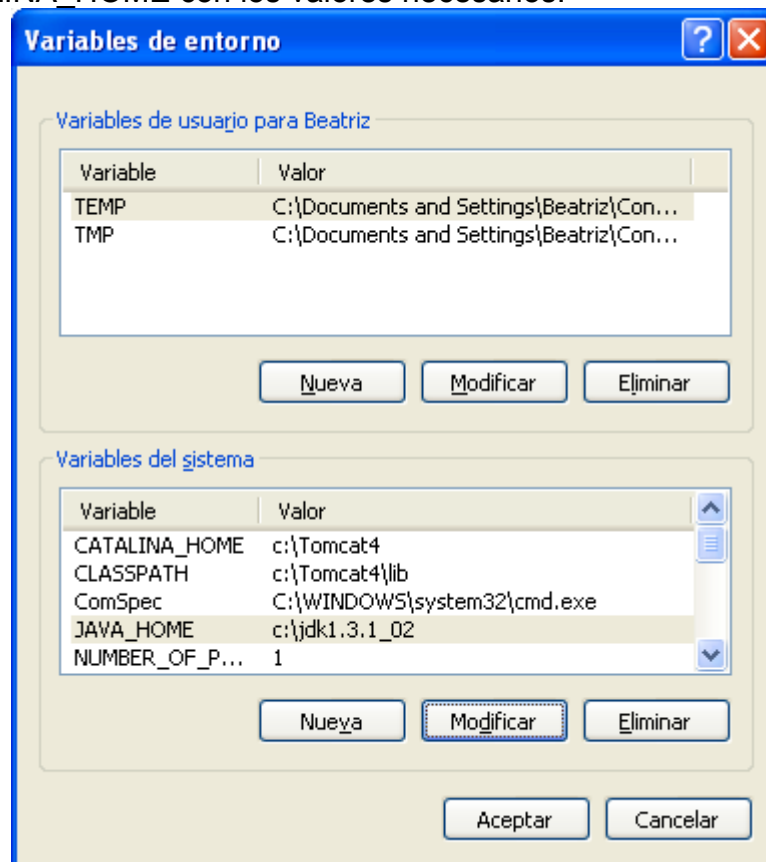
Donde {TOMCAT} es la ruta donde hemos instalado la aplicación.

```
rem ----- Save Environment Variables That May Change -----
set JAVA_HOME=c:\j2sdk1.4.0
set _CATALINA_HOME=c:\tomcat4
```



```
set _CATALINA_BASE=%CATALINA_BASE%
set _CATALINA_HOME=%CATALINA_HOME%
set _CLASSPATH=%CLASSPATH%
set _CP=%CP%
```

- Windows NT/2000/XP: modificaremos las variables de entorno (clic con el botón derecho en *MiPC*, pinchamos en *Propiedades*. Seleccionamos la pestaña *Opciones avanzadas* y pulsamos en el botón *Variables de entorno*). Aquí podremos definir las variables *JAVA_HOME* y *CATALINA_HOME* con los valores necesarios.



Ahora vamos a probar que Tomcat funciona correctamente por sí solo:

1. Para Apache (si se estaba ejecutando).
2. Ejecuta "{TOMCAT}\bin\startup.bat" Tomcat debería iniciarse en una ventana MS-DOS.
3. Abre Internet Explorer e intenta acceder a la dirección <http://localhost:8080/> Deberías ver la página de inicio de Tomcat que además contiene links hacia páginas de ejemplos de servlets y jsp's que podrás ejecutar.

POSIBLE ERROR: Si cuando ejecutas el paso 2, aparece un mensaje que dice "Sin espacio en entorno" deberás ejecutar los siguientes pasos:

1. Accede a las propiedades del fichero "{TOMCAT}\bin\startup.bat" (botón derecho -> propiedades).



2. Dentro de la pestaña *Memoria*, en el apartado *Memoria convencional*, fijar el *Entorno inicial* a 3072 (este valor debería ser suficiente). Si tenemos muy sobrecargado el sistema con variables de entorno podemos aumentar este valor hasta que no nos aparezca el mensaje de "Sin espacio en entorno".
3. Intenta volver a ejecutar tomcat. Esta vez debería funcionar correctamente.

Para parar Tomcat deberás ejecutar "{TOMCAT}\bin\shutdown.bat".

Los ficheros que muestra Tomcat se encuentran dentro del directorio {TOMCAT}\webapps.

3.4. TOMCAT COMO SERVICIO

Copiar el archivo JK_NT_SERVICE.EXE del CD del proyecto (también se puede descargar de la página Web de Tomcat <http://jakarta.apache.org>.) al directorio {TOMCAT}\bin. Dentro del mismo directorio copiar el archivo INSTALAR.JS.

Copiar el archivo WRAPPER.PROPERTIES del CD del proyecto (también se puede descargar de la página Web de Tomcat <http://jakarta.apache.org>.) al directorio {TOMCAT}\conf. Dentro de éste archivo debemos escribir lo siguiente en los lugares correspondientes (asegurarse de que están cambiadas todas las rutas):

```
wrapper.tomcat_home=C:\tomcat4  
wrapper.java_home=c:\jdk1.3.1_02
```

Hay que instalar dos servicios:

- Clasevirtual: Abrir la ventana de comando (MS-DOS). Entrar al directorio {TOMCAT}\bin y ejecutar:

```
jk_nt_service -I clasevirtual c:\tomcat4\conf\wrapper.properties
```


(si no te crea el servicio prueba a ejecutar: *cscript instalar.js clasevirtual*)
- Tomcat4: Modificar el archivo INSTALAR.JS, cambiando las rutas del Tomcat y el JDK. Abrir la ventana de comando (MS-DOS). Entrar al directorio {TOMCAT}\bin y ejecutar:

```
cscript instalar.js tomcat4
```

Si queremos borrar un servicio deberemos realizar lo siguiente:

```
jk_nt_service -R <name of service>
```

En nuestro caso sería:

```
jk_nt_service -R clasevirtual
```



3.5. COMUNICACIÓN APACHE – TOMCAT

En estos momentos tenemos Apache y Tomcat funcionando independientemente. Sin embargo, lo que nosotros queremos es que Apache sirva el contenido estático mientras que Tomcat tiene que ejecutar todos los servlets y todas las páginas jsp que se pidan. Para implementar esta comunicación entre Apache y Tomcat hay varias opciones, en nuestro caso hemos optado por la más moderna y la que se supone es más eficiente: el modulo mod_jk.

Veamos los pasos que tenemos que seguir para instalar este módulo.

1. Copiar “mod_jk.dll” en {APACHE}\modules. Esta dll se puede obtener del sitio Web de tomcat o del CD del proyecto.
2. Modificar el fichero {APACHE}\conf\httpd.conf para incluir la línea “include c:/tomcat4/conf/mod_jk.conf”. Hay que prestar atención a las barras (/) en el path. Esta línea se puede incluir al final del fichero (copiar el workers.properties en {TOMCAT}/conf/, cambiar también las rutas del jdk).
3. Copiar “mod_jk.conf” en {TOMCAT}/conf/. Este archivo se puede obtener del CD del proyecto (asegúrate de cambiar las rutas).

Para probar que mod_jk está funcionando correctamente:

1. Inicia Tomcat y después Apache.
2. Abre una ventana de Internet Explorer y accede a la dirección <http://localhost/examples/servlets/>. Debería aparecer la página de ejemplos de servlets de Tomcat. Fíjate en que esta vez no hemos escritos el puerto 8080 en la dirección, esto se debe a que la petición se la estamos realizando directamente a Apache y es el propio Apache el que se encarga de pedir la petición a Tomcat por medio del módulo recién instalado y servirnosla de un modo correcto.

3.6. LIBRERÍAS XALAN Y XERCES (JAXP 1.2.01)

Esta librería es la que se ha elegido para el tratamiento de los documentos XML mediante DOM y SAX. Usaremos las últimas versiones del interfaz Jaxp 1.2.01

Para instalarla solo tenemos que copiar el archivo xerces.jar, xalan.jar y jaxp.jar dentro del directorio de librerías de Tomcat {TOMAT}\common\lib. Estos archivos se encuentran en el CD del proyecto.

El interfaz jaxp1.2.01 lo podemos encontrar en la página oficial de Sun <http://java.sun.com>. También hay más informaciones sobre Xalan y Xerces en <http://xml.apache.org>.



3.7. PAQUETE COM.OREILLY.SERVLET

Este paquete nos permitirá gestionar la subida de ficheros al servidor sin necesidad de instalar un servidor ftp. Ha sido creada por Jasón Hunter y a día de hoy se puede descargar su última versión de la Web www.servlets.com.

Si miramos su contenido encontraremos una librería llamada cos.jar que será la que nos permita subir ficheros desde nuestra máquina al servidor de la aplicación. Debemos copiar cos.jar en {TOMCAT}\common\lib. Este archivo se encuentra en el CD del proyecto.

3.8. JAVAMAIL 1.2

El JavaMail es necesario para poder mandar correos a los usuarios desde la aplicación.

Basta con copiar el archivo MAIL.JAR en el directorio {JDK}\jre\lib\ext. Este archivo se encuentra en el CD del proyecto. Con esto conseguimos no tener que añadir el jar al classpath, ya que el jdk añade este directorio por defecto al classpath.

Para que funcione correctamente también necesitamos instalar el Java Activation Framework 1.0.1 (JAF). Para ello solo tenemos que copiar el archivo ACTIVATION.JAR en el mismo directorio que hemos copiado el mail.jar. Este archivo se encuentra en el CD del proyecto. Estos ficheros los podemos encontrar también en <http://java.sun.com>.

3.9. FICHERO DE CONFIGURACIÓN 'CONF.TXT' Y 'FTP.TXT'

Estos dos ficheros se encargan de la configuración del sistema. Han de colocarse dentro de: {TOMCAT}\bin\clasevirtual. Estos archivos se encuentran en el CD del proyecto. Estos ficheros son:

- CONF.TXT: como primera línea incluye la ruta del directorio donde está instalada la universidad virtual. Las siguientes líneas indican las direcciones de correo electrónico de cada curso. Cada curso tendrá una dirección de e-mail que será a la cual el sistema envíe las respuestas de los alumnos de las preguntas de tipo texto libre. El formato que han de seguir es el siguiente: primero el identificador del curso, en la siguiente línea la dirección del servidor smtp de correo y en la siguiente línea, la dirección de destino.

Un ejemplo del fichero:

```
c:/tomcat4/webapps/
ims
mailhost.terra.es
rafo@terra.es
curso4
mailhost.terra.es
rafo@terra.es
```

- FTP.TXT: incluye el path al que los profesores envían los cursos por ftp.



3.10. CLASEVIRTUAL

Copiar todo el contenido de la carpeta CLASEVIRTUAL/BUILD (que se encuentra comprimido en el archivo CLASEVIRTUAL.ZIP del CD del proyecto) en el directorio {TOMCAT}\Webapps\clasevirtual.

Arrancar la aplicación

IMPORTANTE:

El servicio clasevirtual debe estar configurado como manual.

El servicio tomcat4 debe estar configurado como manual.

El orden a seguir debe ser el indicado a continuación:

- Iniciar Tomcat: {TOMCAT}\bin\startup.bat (es aconsejable tener un acceso directo a este archivo en el escritorio).
- Iniciar Apache: Inicio->Programas ->Apache HTTP Server->Control Apache Server->Start.
- Abrir el explorador de Windows y poner la dirección: <http://127.0.0.1/clasevirtual/index.html>, con lo que ya debe aparecer la página principal del proyecto.

Finalizar la aplicación

- Cerrar Internet Explorer.
- Parar Apache: Inicio->Programas ->Apache HTTP Server->Control Apache Server->Stop. (Si esta opción no esta disponible prueba a cerrarlo directamente o también puedes probar CONTROL C)
- Para Tomcat, ejecutando: {TOMCAT}\bin\shutdown.bat (es aconsejable tener un acceso directo a este archivo en el escritorio). Conviene pararle tras haber parado Apache.

3.11. POSIBLES PROBLEMAS Y SOLUCIONES

3.11.1. Orden de arranque

Es conveniente arrancar Tomcat en primer lugar y posteriormente arrancar Apache. Si paramos Tomcat y cambiamos su configuración, Apache no la registrará. De modo que lo más conveniente es arrancar Tomcat y después Apache para que el Apache (que es el que recibe todas las peticiones y luego las filtra) tenga registrada la última configuración de Tomcat.

3.11.2. Actualizaciones software

Se supone que el proyecto debería funcionar con versiones superiores de Tomcat, Apache y Java. Aunque bien es sabido que la compatibilidad hacia atrás en software no se puede garantizar al 100%. En nuestro caso tuvimos que cambiar algunos métodos del código en Java al pasar del jdk1.2.1 al jdk1.4 debido a que algunos métodos eran calificados como 'deprecated' por el jdk1.4 y esto puede dar problemas si se usa alguna herramienta de compilación.



3.11.3. Configuración de Apache

Hay que tener cierto cuidado con las líneas con las que rellenamos el fichero httpd.conf de Apache. Si escribimos incorrectamente alguna de estas líneas, es posible que el Apache ni siquiera consiga arrancar. No está de más cerciorarse de que las modificaciones que hay que hacer a este fichero son correctas.

3.11.4. Logs

Muchas veces nos surgirán errores inesperados en Tomcat y Apache. Es de bastante ayuda consultar los logs que nos generan estos programas y en los que aparecen registrados los errores.

En Apache están en: {APACHE}\logs y en Tomcat se encuentran en {TOMCAT}\logs

3.11.5. Classpath

Es muy conveniente tener todas las librerías que vamos a usar en el CLASSPATH del sistema. Más adelante veremos que algunas de las librerías de Java que vamos a añadir, lo haremos en directorios que ya están en el CLASSPATH del jdk para no tener que hacer modificaciones. También es muy importante añadir todas las librerías de Tomcat para compilar. Esto podemos realizarlo añadiendo la siguiente línea en el autoexec.bat (o en las variables de entorno):

```
set CLASSPATH=%CLASSPATH%;c:\tomcat4\lib
```

Suponiendo que c:\tomcat4 sea el directorio en el que hemos alojado el tomcat.

3.12. DIRECCIONES DE INTERÉS

<http://apache.org> Apache Software Foundation

<http://xml.apache.org> Página para el soporte XML bajo Apache.

<http://jakarta.apache.org/> Proyecto Jakarta (aquí podemos encontrar Tomcat).

<http://java.sun.com> Página oficial de Java de Sun Microsystems.

www.servlets.com Utilidades y manuales sobre servlets (aquí podemos encontrar el paquete com.oreilly.servlet con diversas utilidades para los servlets).



4. INTERFAZ GRÁFICA DE USUARIO

4.1. INTRODUCCIÓN

Teniendo en cuenta que estamos inmersos dentro de una sociedad en la que la comunidad de Internet se ha desarrollado hasta tal nivel que ofrece páginas excesivamente variadas y atractivas, se ha considerado que hay que conceder más importancia a una buena presentación que sea llamativa y actual, a la vez que sencilla y legible, que invite al navegante a entrar y a disfrutar con los cursos y que, junto a un contenido interesante y robusto, haga que un visitante ocasional se convierta en un alumno de la clase virtual.

En consecuencia, la presentación de la aplicación ha sido modificada para mostrar una vista más actualizada y dinámica, mejorando el diseño anterior que era poco atractivo, no uniforme y que dificultaba el aprendizaje de los alumnos al desviar su atención.

Las modificaciones han consistido fundamentalmente en el cambio a colores más atractivos y la adición de botones y menús que facilitan la navegación.

4.2. MODIFICACIONES

Las primeras modificaciones que se pueden observar al entrar en el sistema han sido las realizadas en la página de inicio, que se ha rediseñado por completo, agregando un menú flotante que permita el acceso al curso desde cualquier ubicación en la que se encuentre el visitante dentro de la página, además de la inclusión de un nuevo formato para el título, y un bloque de texto deslizante que permite la lectura de una pequeña introducción al curso, sin tener que desplazarse por la página. Aquí hay que aclarar que se han diseñado otras dos páginas de inicio que se adjuntan con el resto de la aplicación y que, si se quiere, podrán ser utilizadas sin mas que sustituir su nombre por el de index.html. Estas páginas se encuentran en la carpeta ../clasevirtual con los nombres index2.html e index3.html.

Para unificar el resto de la aplicación a los cambios realizados en la página de inicio, se ha empleado la misma paleta de colores en todo el sistema. Hemos escogido una tonalidad de azul menos agresiva que el color verde de la aplicación anterior, para facilitar la concentración del alumno, tal y como se ha indicado en la introducción. Para esta decisión se han tenido en cuenta los comentarios o consejos publicados en la Web: www.maestrosdelweb.com por Edgar González Ramos, Arquitecto y Publicista gráfico, que afirma que “los colores son una de las partes más importantes de tu sitio, más que las imágenes que coloques, los colores son los que van a producir sensaciones en la mente de tus visitantes, una buena combinación de colores puede hablar mejor que muchas imágenes” y junto a esta afirmación comenta que un sitio Web debe usar una gama de entre 4 y 8 colores máximo, y que las sensaciones que provocan los distintos colores son:



- **Negro** - Prohibición, Misterio, Muerte, Oscuridad, Maldad.
- **Blanco** - Pureza, Sinceridad, Seriedad.
- **Gris** - Encierro, Aburrimiento, Depresión, Tranquilidad
- **Café** - Monotonía. Tranquilidad, Seriedad.
- **Azul** - Limpieza, Frescura, Tranquilidad, Paz.
- **Amarillo** - Alegría, Calor, Luz, Alerta.
- **Rojo** - Pasión, Fuerza, Agresividad, Alerta.
- **Verde** - Frescura, Naturaleza, Tranquilidad, Indecisión
- **Naranja** - Calor, Hambre, Intranquilidad.
- **Morado** - Misterio, Soledad, Cursi.

Así como también afirma que hay colores que se asocian fácilmente con determinados sitios Web:

- **Negro** - Paginas pornográficas, de contenido ilegal, de videojuegos.
- **Blanco** - Paginas comerciales, de servicios.
- **Gris** - Paginas con contenido muy serio.
- **Café** - Religión, Libros, Coleccionistas y servicios.
- **Azul** - Usado en todas sus tonalidades en casi todos los sitios.
- **Amarillo** - Paginas de chistes, de artesanías, de restaurantes o servicios turísticos.
- **Rojo** - Paginas de servicios generalmente mecánicos, electrónicos o de herramientas.
- **Verde** - Salud, vida sana, nutrición, naturismo.
- **Naranja** - Alimentos, Mascotas.
- **Morado** - Belleza, Eróticas.

Por todos estos consejos, se decidió usar la tonalidad azul en nuestra Web, ofreciendo así un sitio que transmite confianza y tranquilidad a los estudiantes a la vez que se le añade un toque de seriedad con la adición de tonos grises. Todo esto sin sobrepasar en ningún momento de 8 tonos o colores.

Otro de los cambios que cabría destacar de entre los realizados, es la introducción en múltiples páginas del sistema, de un botón o enlace que permite regresar a la página principal, permitiendo así al alumno abandonar el curso desde cualquier ubicación en la que se encuentre.

Se han añadido tanto en el índice de temas a estudiar en cada curso, como en el índice de alumnos para el seguimiento por parte del profesor, sendas cajas con links dinámicos, cuya ventaja es la de permanecer ocultas hasta que no sean necesarias, mejorando así la visibilidad de la página, sin perder nada de funcionalidad, ya que se han mantenido e incluso ampliado las opciones que aparecían en la versión anterior.



En el [Apéndice A](#) se muestra una comparativa entre ambas interfaces. Siguiendo con el propósito planteado en la introducción de facilitar al máximo la lectura y seguimiento de los cursos a los alumnos, se ha modificado la hoja de estilos y las DTD para permitir un formato de texto más amplio. Estos cambios permiten la inclusión de textos formateados tales como: color, negrita, cursiva, subrayado, texto de tamaño distinto, inclusión de listas con viñetas (normales o numeradas)... además de la posibilidad de incluir enlaces a ficheros o direcciones de Internet. De tal manera que los cursos aparecen mejor ordenados con la utilización de listas y viñetas, y más completos al permitir la descarga de elementos complementarios del curso por parte de los alumnos. El que pueda aparecer la negrita, el subrayado, la cursiva... en los diferentes cursos, ayuda a fijar la atención del visitante en aquellos puntos que se crean interesantes.

En el [Apéndice B](#) se muestran los cambios realizados en la implementación.



5. CURSO IMS

Para la realización del curso de IMS se han resumido todos los estándares de IMS, obtenidos de la página principal de esta organización: www.imsglobal.org. Estos estándares y sus versiones han sido los siguientes:

- Content Packaging Specification v1.1.2
- Learning Resource Meta-Data Specification v1.2.1
- Enterprise Specification v1.1
- Question & Test Specification v1.2
- Learner Information Packaging v.1.0
- Learning Design Specification v.1.0
- Digital Repositories Interoperability Specification v.1.0
- Reusable Definition of Competency or Educational Objective Specification v.1.0
- Simple Sequencing Specification v.1.0

Cada uno de estos estándares está a su vez formado por tres documentos bastante extensos, todos de gran complejidad sintáctica y semántica, por lo que aunque se ha resumido todo el contenido lo más posible, la complejidad del curso y sobre todo de algunos de estos estándares, sigue siendo muy compleja.

El curso se ha estructurado en tres partes, la primera consiste en una introducción a los estándares en general y más concretamente a los de IMS. La segunda parte, está formada por el estándar Content Packaging considerado como un estándar básico referenciado por el resto de estándares. Una tercera parte, que profundiza en el resto de los estándares IMS, y que está separada de las primeras partes por un examen, siendo este indispensable para poder acceder a esta última parte.

En el [Apéndice C](#) se puede ver el curso completo. Para ver el curso en condiciones óptimas, se aconseja seguirle a través de la Web, donde se podrán realizar los exámenes y acceder a contenido adicional. Ya que el contenido de dicho curso ha sido diseñado para ser seguido por Internet, puede que la visualización de las imágenes o el propio contenido del curso queden borrosos si se siguen en la versión escrita.



6. ACTUALIZACIÓN AL ESTÁNDAR IMS LEARNING RESOURCE METADATA V1.2.2

Para actualizar el estándar IMS Learning Resource Metadata de la versión 1.2.1 (que es la que se aplicaba en la versión original) hacia la versión 1.2.2 se han realizado los siguientes cambios.

- El namespace XML es: http://www.imsglobal.org/xsd/imsmd_v1p2
- El esquema XML es: http://www.imsglobal.org/xsd/imsmd_v1p2p2.xsd
- Se ha modificado la declaración para el namespace W3C 'XML' de: `xmlns:xml=http://www.w3.org/XML/1998/namespace` a `xmlns:x=http://www.w3.org/XML/1998/namespace`. Esto implica el cambio de las referencias `xml:lang` por `x:lang`.

Se cambian las referencias `##any` por `##other`. Este cambio no afecta a la semántica del esquema y permite que el esquema sea procesado por parsers determinísticos (como Turbo XML y MSXML).

Con estos cambios, se pasa a la versión 1.2.2

- Por tanto, se han revisado los ficheros XML para actualizar las llamadas al esquema.
- También se ha revisado el sistema modificando las llamadas a elementos `<langstring>` que establezcan o lean el atributo `xml:lang`.

Según el documento "IMS Schema Updates" incluido en los archivos del paquete *metadatav1p2p2ex.zip*, no es necesario un cambio en el namespace si no cambia el modelo de información de la especificación o la semántica. Por tanto, en nuestro caso no sería estrictamente necesario porque ni el modelo de información ni el XML binding han cambiado.



7. METADATOS EN FICHERO EXTERNO

7.1. ESTRUCTURA DE META DATOS EN FICHERO EXTERNO

Los meta datos son una descripción del objeto de aprendizaje que los contiene para permitir la búsqueda de recursos y establecer requisitos sobre cómo usar los elementos y representarlos. Así, los meta datos de un manifiesto describen el curso como un todo (porque el manifiesto contiene todo lo que forma parte de un curso, es decir, si algo no aparece en el manifiesto, es porque no pertenece al curso), los meta datos de un ítem describen a ese ítem concreto, etc.

Debido a que esta descripción del objeto de aprendizaje puede ser bastante extensa, se ha decidido incluirla en un fichero externo. De esta manera, el manifiesto no es tan extenso, lo que aumenta la legibilidad del mismo. Esto es especialmente beneficioso en el caso de manifiestos grandes correspondientes a cursos con muchos elementos de aprendizaje y varias organizaciones de estos, etc.

Otro motivo que afianza esta decisión es el de “separar” los distintos estándares de IMS: Content Packaging y Learning Resource Meta-Data. Por un lado estará el manifiesto (que se rige por el primer estándar) y por otro el fichero de meta datos (que sigue el segundo). Lo que se consigue es evitar tener que usar namespaces para incluir los meta datos dentro del manifiesto (por ejemplo: `<imsmd:lom>`, donde *imsmd:* es el namespace que hay que definir al comienzo del manifiesto para la inclusión de meta datos), con la consiguiente mezcla de ambos estándares.

Esta decisión ha afectado al sistema principalmente en la lectura/escritura del título del curso y de la descripción de una organización dada. También afectaría al tratamiento de los prerrequisitos pero, como se verá más adelante, se ha adoptado una solución conforme con la nueva versión del estándar IMS Content Packaging que evita el uso de meta datos para esta labor.

7.1.1. Meta datos en el sistema anterior

La estructura del sistema anterior obligaba al manifiesto a contener los meta-datos dentro de sí mismo. La forma de incluir los meta-datos no era mediante el uso de namespace, sino mediante la modificación del fichero DTD que da la estructura de los elementos que puede contener el manifiesto. Esto se hizo creando un fichero (curso.dtd) que era el DTD que hay que usar según el estándar Content Packaging (imscp_rootv1p1.dtd) al que se añadió el DTD que hay que usar según el estándar Meta-Data (imsmd_rootv1p2.dtd) con la inclusión de la cabecera *imsmd:* para simular el uso de namespace. Los meta datos eran en realidad elementos añadidos a la DTD del manifiesto (es decir, formaban parte del manifiesto), por lo que realmente no se hacía uso de namespaces.



7.1.2.Meta datos en el sistema actual

La estructura del sistema actual obliga a tener un fichero independiente del manifiesto que contenga los meta-datos. Para hacer uso de este fichero, se hace una extensión del DTD que indica el estándar Content Packaging, según la última versión disponible (imscp_v1p1p3.dtd), para añadir un nuevo elemento dentro del elemento *<metadata>*. Este nuevo elemento se llama *<location>* y contendrá el nombre del citado fichero de meta datos (con la restricción de que debe estar en el mismo directorio que el manifiesto).

El estándar IMS Content Packaging no hace referencia a la implementación de meta datos. Únicamente se dice que las referencias que se hagan a meta datos dentro del manifiesto deben cumplir el estándar IMS Learning Resource Meta-Data. Por tanto la implementación está abierta a las distintas soluciones que pueda dar cada implementador. Se da la posibilidad de añadir extensiones al DTD (imscp_v1p1p3.dtd) de forma que se cree uno nuevo (apartado 7.4 del documento IMS Content Packaging Best Practice Guide Version 1.1.2, página 22) y esta es la solución adoptada.

La decisión de incluir un elemento llamado *<location>* se tomó basándonos en la solución que encontramos en algunos de los ejemplos que ADL SCORM ofrece en su página Web (www.adlnet.org). Aunque el sistema que usa SCORM es distinto, se ha adoptado la inclusión de dicho elemento porque cumple con los requisitos que buscábamos (explicados ya anteriormente).

7.2. IMPLEMENTACIÓN DE META DATOS EN FICHERO EXTERNO

7.2.1.Fichero de meta-datos

El fichero que contiene la meta-dato se ha creado desde cero debido a que no existía en la versión anterior.

Para la realización del fichero de meta datos se ha seguido el estándar Learning Resource Meta-Data v1.2.2.

7.2.1.1. Decisiones tomadas en la creación de dicho fichero

Nota: Se aconseja ir siguiendo las decisiones con el fichero imsmd.xml delante. Como palabras clave (keyword), hemos puesto las palabras relacionadas con los distintos estándares que se incluyen en IMS.

Se considera la estructura lineal porque la forma común de seguir el curso es leer los diferentes documentos unos detrás de otros.

El nivel de agregación es 3 porque es una Web con índice formada por varias páginas independientes, imágenes, ficheros descargables, etc.

Como la aplicación se ha desarrollado bajo sistema operativo Windows, este es su requisito. También es necesario como navegador tener el Microsoft Internet Explorer.

En el apartado *<educational>* se dice que el tipo de interacción es expositivo porque la forma de aprendizaje es mostrar los distintos estándares al alumno, mediante texto narrativo (indicado en el elemento *<learningsourcetype>*). Se ha



considerado el nivel de interacción muy bajo porque el alumno solamente debe leer la teoría y responder a los distintos exámenes que se le plantean. La densidad semántica es alta porque el contenido del curso son los resúmenes de los diferentes estándares.

Se ha considerado la dificultad alta porque el contenido es muy denso y abstracto.

En la primera versión de meta datos se ha considerado que no hay copyright ni coste alguno derivado del uso de la aplicación.

El propósito del curso se ha tomado como objetivo educativo, para que refleje la intención educativa de la clase virtual.

7.2.2.Ficheros imsmanifest.xml y curso.dtd

7.2.2.1. curso.dtd

Se ha mantenido la nomenclatura anterior aunque se ha creado de nuevo. El estándar no obliga a usar ningún nombre en particular para los DTD's extendidos.

Este fichero está formado por el DTD que hay que seguir según el estándar IMS Content Packaging (imscp_v1p1p3.dtd) con la modificación para poder usar los meta datos guardados en un fichero externo (es decir, añadiendo el elemento `<location>` dentro del elemento `<metadata>`).

7.2.2.2. imsmanifest.xml

Las modificaciones realizadas respecto a los meta-datos en este fichero son varias:

1. Se ha eliminado la parte de meta datos del elemento `<metadata>` del manifiesto (elemento `<manifest>`) en la cual se incluía únicamente el título del curso.
2. Se ha añadido en su lugar el elemento `<location>`, que contiene el nombre del fichero que contiene los meta-datos.
3. Se han realizado los puntos 1 y 2 para cada elemento `<organization>`.

7.2.3.Fichero curso.java

La decisión de guardar los meta-datos dentro de un fichero externo ha afectado principalmente a este fichero y más concretamente a los siguientes métodos.

7.2.3.1. Método: titulo()

Este método devuelve el título de un curso, el cual se indica dentro del elemento `<title>` de los meta datos del manifiesto y en caso de no existir dicho elemento, se devuelve el identificador del curso, que es el identificador del elemento `<manifest>` dentro del fichero *imsmanifest.xml*.

En la versión original, el título del curso debía estar en la siguiente ruta dentro del manifiesto: manifest / metadata / imsmid:lom/ imsmid:general / imsms:title / imsmid:langstring.

Con las nuevas versiones se ha cambiado la ubicación del título ya que ahora está en un fichero externo de meta datos (dentro de la ruta:



lom/general/title/langstring) al que se referencia desde un elemento del manifiesto (en la ruta del fichero imsmanifest.xml: manifest/metadata/location).

7.2.3.1.1. Versión original

La versión original de la clase curso.java creaba el árbol del manifiesto, después recorría el fichero buscando elementos que se llamasen *<manifest>* y cogía el primero, para guardar el identificador del manifiesto. Luego vuelve a recorrer el árbol buscando apariciones de cada uno de los elementos (*<metadata>*, *<imsmd:lom>*, *<imsmd:general>*, *<imsmd:title>* y *<imsmd:langstring>*) de la ruta hasta el título. Tras sacar cada elemento, preguntaba si era vacío para devolver como valor de la función el identificador del curso. Cuando comprobaba que *<imsmd:langstring>* no era vacío, entraba en un bucle *for* que recorría todos los hijos de dicho elemento preguntando para cada uno si el tipo de nodo del hijo era texto y distinto de vacío para devolver entonces el valor de dicho hijo (que contendrá el título del curso), en caso contrario se devolvía el identificador del curso.

7.2.3.1.2. Versión 1

La primera versión se hizo usando el mismo método para acceder a un elemento que la versión original, que era recorrer el árbol buscando elementos que se llamen igual que el que estamos buscando y se cogía el primero de ellos.

La modificación de esta versión consiste en buscar el elemento *<manifest>* y extraer su identificador, para acceder a los elementos *<metadata>* y *<location>*, preguntando si son vacíos para devolver el identificador en caso de que lo sean. El valor del hijo de *<location>* se usa para sacar la ubicación del archivo donde están los meta datos. Se crea un nuevo árbol con los elementos que están en el archivo mencionado anteriormente y se recorre para encontrar el título usando la técnica anterior (recorrer el árbol buscando los elementos *<general>* y *<title>*). Si alguno de los elementos intermedios no existe, se devuelve como título el identificador del manifiesto. Guardamos el elemento *<langstring>* (según el estándar IMS Learning Resource Meta-Data sabemos que si existe el elemento *<title>* entonces existe uno o más elementos *<langstring>*) para obtener su hijo (que será único si cumple dicho estándar). Si este hijo no está vacío contendrá el título del curso y se devuelve, y si está vacío se devuelve el identificador del curso.

7.2.3.1.3. Versión 2

Primero se extrae el identificador del manifiesto (que es el atributo del elemento *<manifest>*), después mediante Xpath accedemos al elemento *<location>* directamente y si no existe tal elemento o no existe el elemento *<metadata>*, se devuelve el identificador del curso. Luego obtenemos el nombre del fichero a partir del elemento *<location>* y se crea el árbol de dicho fichero (que será el que contenga los meta datos del curso). Se accede a la posición donde debe ir el título (en la ruta: lom/general/title/langstring) y si es vacío se devuelve el identificador del curso. Cogemos el texto del título y si no es vacío lo devolvemos (en otro caso, si es vacío se devolverá el identificador).



7.2.3.2. Método: pontitulo(titulo)

Este método establece el título de un curso.

Como se ha explicado anteriormente, en la versión original, el título del curso debía estar en los meta datos del manifiesto y con las nuevas versiones en un fichero de meta datos externo.

7.2.3.2.1. Versión actual

Primero se busca el nombre del fichero que contiene los meta datos dentro de la ruta manifest/metadata/location y si no existe este elemento, lo crea, pone por defecto como nombre del fichero *imsmd.xml* y crea dicho fichero. Después abre el archivo de meta datos y si existe el título lo cambia y si no, lo crea.

7.2.3.3. Método: descripcionorganizacion(id)

Devuelve la descripción de la organización. Este método se invoca cuando entra un alumno a un curso y, existiendo dos o más organizaciones del curso, y pulsa en cambiar de organización y se muestran las organizaciones disponibles junto con su descripción. En el caso de que no exista dicha descripción se devuelve la cadena vacía.

En el sistema original la descripción está dentro de los meta datos del elemento *<organization>* y en la última versión se guarda en un fichero externo con los meta datos de dicho elemento.

7.2.3.3.1. Versión actual

Se busca en la organización dada el nombre del fichero, se abre y se busca el título, y si no existe el fichero o el título, se devuelve la cadena vacía.

7.2.3.4. Método: pondescpcionorganizacion (id, descripcion)

Este método establece la descripción de una organización dada.

Como se ha explicado anteriormente, en la versión original, la descripción debía estar en los meta datos de la organización y con la nueva versión en un fichero de meta datos externo.

7.2.3.4.1. Versión actual

Primero se busca el nombre del fichero que contiene los meta datos dentro de la organización y si no existe este elemento (*<location>*), lo crea, pone por defecto como nombre del fichero el identificador de la organización con la extensión *.xml* y crea dicho fichero. Después abre el archivo de meta datos y si existe la descripción lo cambia y si no, la crea.



8. TRATAMIENTO DE LOS PRERREQUISITOS DE UN ÍTEM

8.1. PRERREQUISITOS EN EL SISTEMA ORIGINAL

El sistema original resolvía el problema de los prerrequisitos añadiendo la dependencia del fichero dentro del elemento `<metadata>` del propio `<item>`, indicándolo con el valor "Requires" dentro del elemento `.../item/metadata/imsmd:lom/imsmd:relation/imsmd:kind/imsmd:value/imsmd:langstring`. Así, para indicar que el ítem A tiene como prerrequisito al ítem B (es decir, para poder acceder al ítem A antes hay que haber visitado o realizado el ítem B), se incluía en el ítem A el elemento dentro de la ruta anterior rellenando su valor con el identificador del ítem B.

En el siguiente ejemplo, que se ha tomado del `imsmanifest.xml` del curso `ims` anterior, se puede observar que los prerrequisitos del ítem llamado `ELEM4-1` son los ítems `ELEM2-2` y `EX1`. Así, para poder acceder al ítem `ELEM4-1`, antes se debe haber realizado o visitado los ítems `ELEM2-2` y `EX1`.

8.1.1. Ejemplo de prerrequisitos en el sistema original

...

```
<item identifier="ELEM2-2" identifierref="resource22">
  <title> Prerrequisito</title>
</item>
<item identifier="EX1" identifierref="resource1">
  <title> Examen</title>
</item>
<item identifier="ELEM4-1" identifierref="resource41">
  <title> Introducción</title>
  <metadata>
    <imsmd:lom>
      <imsmd:relation>
        <imsmd:kind>
          <imsmd:source>
            <imsmd:langstring xml:lang="x-none">LOMv1.0</imsmd:langstring>
          </imsmd:source>
          <imsmd:value>
            <imsmd:langstring xml:lang="x-none">Requires</imsmd:langstring>
          </imsmd:value>
        </imsmd:kind>
        <imsmd:resource>
          <imsmd:identifier>ELEM2-2</imsmd:identifier>
        </imsmd:resource>
      </imsmd:relation>
      <imsmd:relation>
        <imsmd:kind>
          <imsmd:source>
            <imsmd:langstring xml:lang="x-none">LOMv1.0</imsmd:langstring>
          </imsmd:source>
          <imsmd:value>
            <imsmd:langstring xml:lang="x-none">Requires</imsmd:langstring>
          </imsmd:value>
        </imsmd:kind>
        <imsmd:resource>
          <imsmd:identifier>EX1</imsmd:identifier>
        </imsmd:resource>
      </imsmd:relation>
    </imsmd:lom>
  </metadata>
</item>
```



```
</imsmd:resource>
</imsmd:relation>
</imsmd:lom>
</metadata>
</item>

...

<resource identifier="resource22" href="22.xml">
  <file>22.xml</file>
</resource>
<resource identifier="resourceee1" href="e1.xml">
  <file>e1.xml</file>
</resource>
<resource identifier="resource41" href="41.xml">
  <file>41.xml</file>
</resource>

...
```

8.2. PRERREQUISITOS EN EL SISTEMA ACTUAL

Para el tratamiento de los prerrequisitos en el sistema actual se ha hecho uso del elemento `<dependency>`, cuya ruta en Xpath es: `manifest/resources/resource/dependency` y por tanto forma parte de los recursos. Esto implica que los prerrequisitos se indican dentro de los recursos asociados a los ítems en lugar de dentro de los propios ítems, es decir, para indicar que el ítem A tiene como prerrequisito al ítem B, se añade el elemento `<dependency>` en el recurso asociado al ítem A con el valor del recurso asociado al ítem B. El elemento `<dependency>` aparece cero o más veces dentro de un recurso y tiene un atributo obligatorio (*identifierref*) que es el identificador del recurso del cual depende.

En el siguiente ejemplo se muestran los ítems del ejemplo anterior aplicando el tratamiento actual para los prerrequisitos. Se puede observar la diferencia en tamaño, ya que se necesitan menos elementos para indicar lo mismo, y en claridad, porque las dependencias se aprecian a nivel de recursos que es menos abstracto que a nivel de ítems (es más fácil relacionar dos ficheros físicos que dos ítems de los que sólo vemos en un primer vistazo los identificadores y los títulos).



8.2.1. Ejemplo de prerrequisitos en el sistema actual

```
...
<item identifier="ELEM2-2" identifierref="resource22">
  <title> Prerrequisito </title>
</item>
<item identifier="EX1" identifierref="resourcee1">
  <title> Examen</title>
</item>
<item identifier="ELEM4-1" identifierref="resource41">
  <title> Introducción</title>
</item>
...
<resource identifier="resource22" href="22.xml">
  <file>22.xml</file>
</resource>
<resource identifier="resourcee1" href="e1.xml">
  <file>e1.xml</file>
</resource>
<resource identifier="resource41" href="41.xml">
  <file>41.xml</file>
  <dependency idenfierref="resource22"/>
  <dependency idenfierref="resourcee1"/>
</resource>
...
```

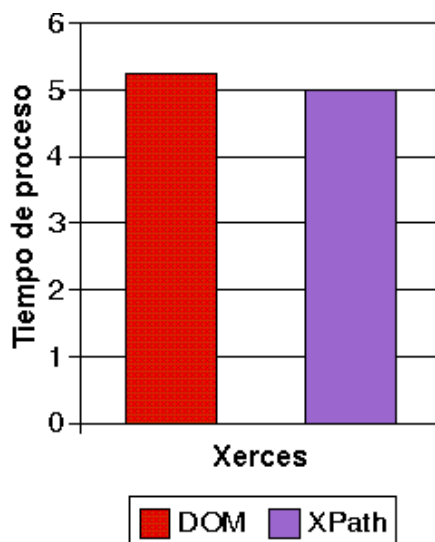
9. SUSTITUCIÓN DE DOM POR XPATH

DOM es un conjunto de interfaces que permite a los programas acceder y modificar dinámicamente el contenido, estructura y estilo de documentos XML. Es independiente del lenguaje y la plataforma. Centrándonos en Java, los objetos DOM se usan para extraer información de documentos XML (empleando el parser XML Apache Xerces). Los elementos tienen relaciones padre-hijo con otros elementos. Con este API, el analizador construye una estructura interna por la que una aplicación puede navegar. DOM permite que una aplicación tenga acceso aleatorio al documento estructurado como un árbol, y el coste es el incremento en la memoria a utilizar.

XPath es un lenguaje para direccionar partes de un documento XML.

Utiliza una sintaxis compacta y no XML para facilitar el uso de XPath dentro de URIs y de valores de atributos XML. Opera sobre la estructura lógica abstracta de un documento XML, más que en su sintaxis superficial. Obtiene su denominación por el uso que hace de una notación de caminos, como en los URLs, para navegar a través de la estructura jerárquica de un documento XML. XPath modela un documento XML como un árbol de nodos. La evaluación de expresiones tiene lugar respecto a un contexto, se invierte la mayoría del tiempo en localizar los elementos relevantes para ser procesados con las expresiones XPath.

La decisión de sustituir DOM por XPath en el acceso a los datos, se toma basándose en que ésta tecnología presenta una mayor simplicidad a la hora de codificar e invierte menos tiempo en procesar los documentos. La tecnología DOM sigue presente en el proyecto ya que XPath sólo permite acceso a datos y no creación, modificación, sustitución... de nodos.



Esta prueba midió el tiempo para procesar documentos que describen 1000, 2000, 3000, 4000 y 5000 configuraciones de tableros de ajedrez (aunque aquí sólo presenta el resultado de los documentos de 1000 configuraciones). Cada documento ha sido procesado 10 veces en 60 ejecuciones. El tiempo medido fue sumado al tiempo del sistema y de usuario, y devuelto por el comando `ptime`, dividido por el número total de documentos procesados.

Comparación de las Tecnologías Java para XML



Un claro ejemplo para demostrarlo, obtenido de la página Web <http://www.programacion.com/tutorial/javaxml/2/> se muestra a continuación:

```
Node nodo_profesor = doc.getFirstChild();
for ( Node hijo_doc = doc.getFirstChild();
      hijo_doc != null;
      hijo_doc = hijo_doc.getNextSibling() ){
    if ( ( hijo_doc.getNodeType() == Node.ELEMENT_NODE ) &&
          ( hijo_doc.getNodeName().equals("profesor") ) ){
        // Guardamos en 'nodo_profesor' el elemento 'profesor'
        nodo_profesor = hijo_doc;
    }
}

// Recorremos los hijos de 'profesor'
for ( Node hijo_profesor = nodo_profesor.getFirstChild();
      hijo_profesor != null;
      hijo_profesor = hijo_profesor.getNextSibling() ){
    // Miramos que el hijo sea de tipo elemento y que tenga por nombre 'cursos'
    if ( ( hijo_profesor.getNodeType() == Node.ELEMENT_NODE ) &&
          ( hijo_profesor.getNodeName().equals("cursos") ) ){
        // Recorremos los hijos de 'cursos' para obtener los identificadores de los cursos
        // que imparte el profesor
        for ( Node hijo_cursos = hijo_profesor.getFirstChild();
              hijo_cursos != null;
              hijo_cursos = hijo_cursos.getNextSibling() ){
            // Miramos que el hijo sea de tipo elemento y que tenga por nombre 'curso'
            if ( ( hijo_cursos.getNodeType() == Node.ELEMENT_NODE ) &&
                  ( hijo_cursos.getNodeName().equals("curso") ) ){
                // Obtenemos el valor del identificador y lo guardamos en el vector 'cursos'
                NamedNodeMap nnm = hijo_cursos.getAttributes();
                // Procesamos los atributos
                for( int i = 0; i < nnm.getLength(); i++ ){
                    Attr atributo = ( Attr ) nnm.item(i);
                    if ( atributo.getName().equals("identificador") ){
                        // El atributo es el identificador del curso
                        String id_curso = atributo.getValue();
                        cursos.add(id_curso);
                    }
                }
            }
        }
    }
}
```



El anterior fragmento de código procesado mediante DOM, se puede sustituir por el siguiente fragmento procesado mediante XPath:

```
NodeList idcursos = null;
try{
    idcursos = XPathAPI.selectNodeList(doc,"./profesor/cursos/curso/@identificador");
}catch(Exception e){};
if (idcursos != null)
    for (int i=0; i<idcursos.getLength(); i++){
        cursos.add(idcursos.item(i).getNodeValue());
    }
```

Como se puede observar, la simplicidad es mucho mayor y a la hora de procesar el documento se sustituyen varios bucles for anidados (seleccionar los profesores, recorrer los hijos para buscar cursos, recorrer los cursos para buscar elementos curso...), que presentan una complejidad considerable, por un acceso simple (directamente devuelve los identificadores de todos los nodos curso) y un solo bucle for, para añadir al vector cursos los resultados.



10. DIRECTRICES FUTURAS: ACTUALIZACIÓN DE PAQUETE DE CONTENIDO A UNIDAD DE APRENDIZAJE SEGÚN EL ESTÁNDAR IMS LEARNING DESIGN

10.1. INTRODUCCIÓN

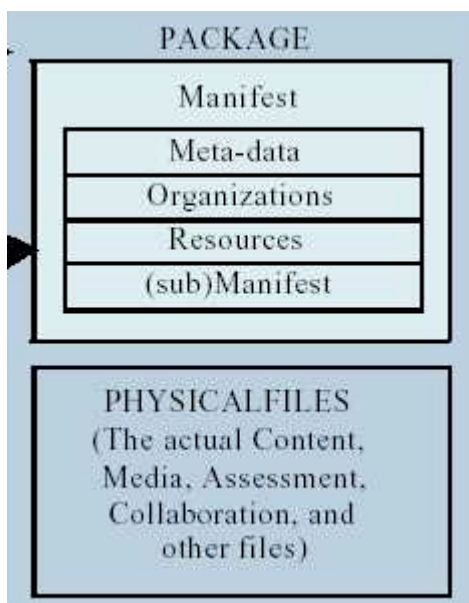
Este estándar apareció, por primera vez, como versión de prueba, en Octubre del 2002 y no fue hasta finales de Febrero 2003, en mitad de nuestro proyecto, cuando apareció su primera versión definitiva. Esto hizo que el estudio y traducción del estándar no estuviera disponible hasta fechas en las que la realización de los cambios que este estándar requiere hubieran podido poner en riesgo la finalización del proyecto en el plazo indicado. Por este motivo, y tras una valoración exhaustiva de los cambios a realizar, se estimó que se dejaría todo el estudio realizado planteado como una opción para versiones posteriores, en las que se pueda tener una visión más amplia sobre la utilización y aplicación del nuevo estándar.

10.2. DESCRIPCIÓN

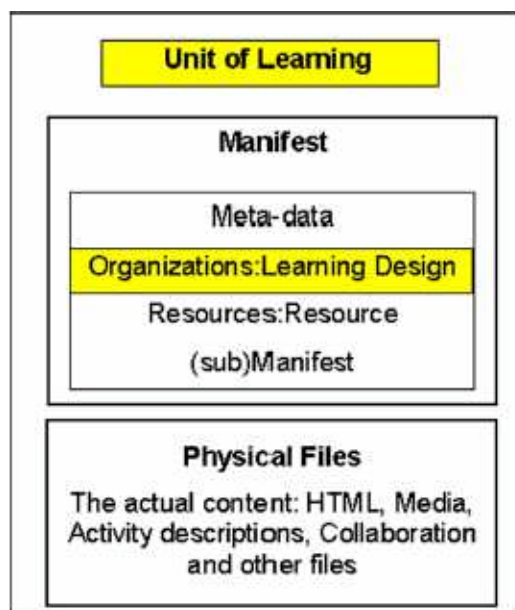
Este estándar ha surgido para dar un entorno de trabajo que tenga los elementos necesarios para describir cualquier diseño de un proceso enseñanza-aprendizaje. El método empleado para conseguir esto se llama **Learning Design** (Diseño de Aprendizaje) y consiste en dar un conjunto de **Actividades** (de resolución de problemas, de búsqueda, de discusión, de valoración, etc.) a los diferentes Roles (o papeles) del alumno, para que las pueda ir completando en un cierto orden y así, ir alcanzando los objetivos de aprendizaje propuestos en dicho proceso enseñanza-aprendizaje.

10.3. COMPARACIÓN CON EL IMS CONTENT PACKAGING Y OTROS ESTÁNDARES

Este estándar es muy parecido en cuanto a objetivos a la especificación **IMS Content Packaging (Paquete de Contenido IMS)** que también describe estructuras de datos que se utilizan para desarrollar un proceso enseñanza-aprendizaje en Internet y así permitir el intercambio de información entre aplicaciones que usen el mismo estándar. Pero hay muchas diferencias estructurales entre ambos que hacen que **no** sean compatibles entre sí, ya que, el principal uso del Diseño de Aprendizaje es crear o modelar Unidades de aprendizaje, que no son más que Paquetes de Contenido modificados a los que se añade un elemento nuevo *<learning design>* dentro del elemento *<organizations>* que se encuentra en el Manifiesto y que por lo tanto, pasan de ser un **Paquete de Contenido IMS** a una **Unidad de Aprendizaje**, como se ve en la siguiente figura:



Paquete de Contenido IMS



Unidad de Aprendizaje

A parte de este cambio, habría que quitar todas las demás organizaciones que se encontraran en el elemento *<organizations>* y colocarlas en submanifiestos, si no, si las dejáramos dentro del elemento *<organizations>* junto al nuevo elemento incorporado *<learning design>*, serían ignoradas. Estos cambios se ven en el cuadro comparativo expuesto más abajo.

Las especificaciones (estándares) IMS hasta esta última, reflejan un modelo para un sólo usuario, un estudiante, que interactúa con los contenidos para ir aprendiendo y que puede ser evaluado de esos conocimientos que va adquiriendo.

El Learning Design (Diseño de Aprendizaje) nivel A da la capacidad de diseñar Unidades de Aprendizaje que incluyen simultáneamente diversos roles o papeles, cada uno de los cuales puede ser interpretado por varios actores. Esto permite que las Actividades correspondientes sean especificadas en “flujos de aprendizaje” coordinados que son análogos para flujos con objetos del mismo tipo o grupo. De esta manera, esta especificación soporta aprendizaje en grupo y de colaboración de muy diferentes tipos, cuya importancia está cada vez más reconocida en las esferas educacionales y en la formación comercial. Esta especificación también puede ser usada para soportar un modelo de un solo estudiante mediante la creación de una Unidad de Aprendizaje con un único rol definido y ninguna interacción entre los estudiantes. Además de que el Diseño de Aprendizaje (Learning Design) permite una mayor reutilización de los componentes ya que se separan las Actividades, de las **Estructuras de Actividades** (grupos de actividades que aportan la ventaja de hacer que las diferentes actividades en grupos puedan ser opcionales y no siempre secuenciales como lo eran antes, ver cuadro comparativo más abajo: *<activity-structure identifier=" " number-to-select="2" structure-type=" selection sequence">* reúne dos actividades y permite seleccionar una de ellas sin que tengan que ir una detrás de la otra), de los recursos y de los roles y que luego



se reúnen en un **Método** que se encarga de interrelacionarlas todas entre sí (es como una obra con diferentes actores en cada acto).

El nivel B añade una mayor personalización de los estudiantes, soportando conocimiento anterior al curso, accesibilidad, preferencias... Facilitando así, por ejemplo, la definición de entornos de trabajo, objetivos y prerequisites que permitirían una mayor individualización en las tareas de cada alumno.

El nivel C añade la capacidad de envío de mensajes entre los diferentes componentes del sistema y entre roles. Esto añade una nueva dimensión ya que permite diferentes flujos de educación conducidos por eventos y en tiempo real.

10.4. USO EN NUESTRA APLICACIÓN.

Si bien el uso de este estándar reportaría mejoras a la aplicación como se ha explicado antes, todos los cambios mencionados en el apartado anterior y el hecho de que el estándar haya sido una Versión 1.0 Public Draft Specification (versión de prueba, no definitiva) hasta finales de Febrero del 2003, hacen que el uso de este estándar en nuestra aplicación conlleve un gasto en tiempo y complejidad considerable, que a priori no parece compensar, por lo menos hasta que el estándar no se encuentre más extendido y avanzado, ya que nuestra aplicación se basa en recorridos por los elementos del documento `imsmanifest.xml` que es el que da la estructura del curso a seguir, y si esto se modifica, se deberían modificar todos los procedimientos de búsqueda de dichos elementos que tiene la aplicación, cambios estudiados con detenimiento en el [Apéndice D](#), cambios que se deberían añadir a los ya mencionados en apartados anteriores correspondientes a adaptar el Manifiesto al estándar.

10.5. CUADRO COMPARATIVO ENTRE EL IMSMANIFES.XML EN UN PAQUETE DE CONTENIDO Y EL IMSMANIFEST.XML EN UNA UNIDAD DE APRENDIZAJE

<pre><manifest identifier="ims" xmlns:imsmd="imsmd.xml"> <metadata> <schema>IMS Content</schema> <schemaversion>1.1</schemaversion> <location>imsmd.xml</location> </metadata></pre>	<pre><manifest xmlns=" " xmlns:imsId=" " xsi:schemaLocation> <metadata> <schema>IMS Metadata</schema> <schemaversion>1.2</schemaversion> <imsmd:general> <imsmd:title> Anticipating</imsmd:title> </imsmd:general> </metadata></pre>
<pre><organizations default="indiceCurso"> <organization identifier="indiceCurso"> <title>Indice del Curso de IMS</title> <item identifier="E1" identifierref="r1"> <title> titulo </title> </item> <item identifier=" " identifierref=" "> <title> </title> <item identifier="" identifierref=" "> <title></title> </item> --RESTO DE ITEMS DE DENTRO </item> <item identifier="EX1" identifierref=" "></pre>	<pre><organizations> <learning-design identifier=" " uri=" " level="A"> <title> </title> <learning-objectives> <item identifierref="" identifier=" "/> </learning-objectives> <prerequisites> <item identifierref="" identifier=""/> </prerequisites> <components> <roles> <learner identifier="learner"/> </roles> <activities> <learning-activity identifier=" "></pre>

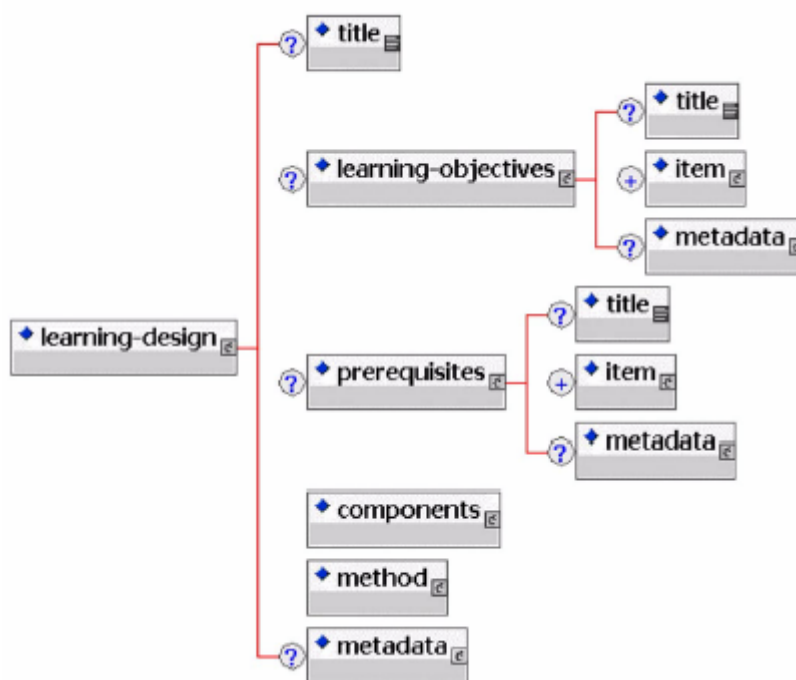


<pre><title>Examen </title> </item> <item identifier=" "> <title> </title> <item identifier="" identifierref=" "> <title></title> <metadata> <imsmd:lom> <imsmd:relation> <imsmd:kind> <imsmd:source> --RESTO DE METADATOS </imsmd:lom> </metadata> </item> ---- RESTO DE ÍTEMS </organization> </organizations></pre>	<pre><activity-description> <item identifierref="" identifier=""/> </activity-description> </learning-activity> ---RESTO DE ACTIVIDADES <activity-structure identifier=" " number-to- select="2" structure-type="sequence"> <title/> <learning-activity-ref ref=" "/> <learning-activity-ref ref=" "/> </activity-structure> <activity-structure identifier=" " number-to- select="2" structure-type="selection"> <title/> <learning-activity-ref ref=" "/> <learning-activity-ref ref=" "/> </activity-structure> ---RESTO DE ESTRUCTURAS </activities> <environments> <environment identifier="I"> <title> I</title> </environment> </environments> </components> <method> <play identifier="" isvisible=" "> <act identifier=" "> <role-part identifier=" "> <role-ref ref="learner"/> <activity-structure-ref ref=""/> </role-part> <complete-act> <when-role-part-completed ref=" "/> </complete-act> </act> <complete-play> <when-last-act-completed/> </complete-play> </play> </method> </learning-design> </organizations></pre>
<pre><resources> <resource identifier="r1" type="eaula" href=" "> <file href="e1.xml"/> </resource> -----RESTO DE RECURSOS </resources></pre>	<pre><resources> <resource identifier="r1" type="eaula" href=" "> <file href="e1.xml"/> </resource> -----RESTO DE RECURSOS </resources></pre>
<pre></manifest></pre>	<pre></manifest></pre>
Imsmanifes.xml en un Paquete de Contenido	Imsmanifest.xml en una Unidad de Aprendizaje

El elemento `<location>` es distinto porque se ha añadido este año para poder acceder a los meta-datos que se han colocado en un fichero fuera del `imsmanifest.xml`.

10.6. RELACIÓN DEL ESTÁNDAR CON LOS META-DATOS

Como ya se ha mencionado en otros estándares, se pueden usar los meta-datos para referenciar un (Learning Object) **Objeto de Aprendizaje** cualquiera del estándar, que es cualquier entidad digital o no, que pueda ser usada, rehusada o referenciada durante el aprendizaje (con tecnologías), por ejemplo, material impreso, temas a estudiar, ejercicios, textos, cursos, programas para estudiar e incluso personas. En general suelen ser elementos más pequeños que los cursos y que pueden ser rehusados en diferentes cursos, que en su mayoría no pueden ser rehusados ni intercambiados. Estos Objetos de Aprendizaje pueden ser entidades a las que se accede con Meta-Datos, que se guardan de forma separada a los Objetos de Aprendizaje a los que se refieren, por ejemplo en Bases de Datos. Una imagen de un ejemplo de como se integran los Meta-Data en este estándar, aparece en la siguiente figura:



Donde cada vez que aparece el recuadro *metadata*, significa que ahí se incluyen los Meta-Data IMS correspondientes siguiendo el estándar y mediante el namespace correspondiente. El elemento `<metadata>` es opcional y puede aparecer cero, una o más veces como se explica a continuación.

Dentro del estándar IMS Learning Design se proporciona un mecanismo para la inclusión de meta datos dentro de los diferentes elementos del paquete. Así, es posible añadir meta datos para describir a los diferentes elementos que forman parte de una unidad de aprendizaje: manifiesto, objetivos de aprendizaje, prerequisites, componentes, métodos.



Al igual que en el IMS Content Packaging, el meta dato siempre describe al elemento dentro del cual está incluido. Así, el meta dato del manifiesto describe la Unidad de Aprendizaje como un todo, y así el de cada elemento del árbol donde aparece un elemento *<metadata>* (la figura está incompleta, el árbol de elementos es mucho mayor). Hay que tener en cuenta que si el elemento tiene (sub)elementos, el meta dato describirá al elemento y los (sub)elementos como un todo.

10.6.1. Estructura de los meta datos

En el estándar IMS Learning Design hay dos elementos *<metadata>* diferenciados: el del manifiesto y el de los demás (sub)elementos al igual que ocurría en el Content Packaging.

10.6.1.1. <metadata> dentro de <manifest>

Aparece cero o una vez dentro del elemento *<manifest>*.
Elementos:

10.6.1.1.1. <schema>

Describe el esquema usado (por ejemplo: IMS Content, ADL Content, etc.). Si no aparece este elemento, se supone que es "IMS Content". Este elemento aparece cero o una vez.

No tiene atributos ni elementos.

Ejemplo: *<schema>IMS Content</schema>*

10.6.1.1.2. <schemaversion>

Dice la versión del esquema anterior. Si no aparece la versión, se asume que es "1.1". Este elemento aparece cero o una vez.

No tiene atributos ni elementos.

Ejemplo: *<schemaversion>1.1</schemaversion>*

10.6.1.1.3. IMS Meta-Data

Se puede añadir cualquier elemento definido en la especificación IMS Meta-Data. El número de veces que puede aparecer este elemento depende del estándar IMS Meta-Data.

10.6.1.2. <metadata> dentro del resto de elementos:

Aparece cero o una vez dentro del elemento que lo contenga.

Se puede añadir cualquier elemento definido en la especificación IMS Meta-Data. El número de veces que puede aparecer este elemento depende del estándar IMS Meta-Data.



11. ARQUITECTURA Y ENTORNO TECNOLÓGICO

La arquitectura general elegida se basa en una “arquitectura de tres capas”, según la actual terminología de ingeniería del software. Estas capas son las de Presentación, Negocio y Datos.

Lo que esta arquitectura implica es la separación del software del sistema en tres bloques en función del tipo de servicio que proporciona cada uno de ellos. Así:

- La capa de **Datos**: contendrá el software que proporciona las funciones de acceso a los datos del sistema. Esta capa es responsable de:
 - Almacenar datos.
 - Recuperar datos.
 - Mantener los datos.
 - Integridad de datos.
- La capa del **Negocio**: se encargará de automatizar las reglas del negocio propias del sistema. Los servicios de negocios son el “puente” entre los servicios de presentación y los servicios de datos. Esto aísla al usuario de la interacción directa con la base de datos. Las reglas de negocio son políticas que controlan el flujo de las tareas. La capa de negocio es responsable de:
 - Recibir la entrada de la capa de presentación.
 - Interactuar con los servicios de datos para ejecutar las operaciones de negocios para los que la aplicación fue diseñada.
 - Enviar el resultado procesado a la capa de presentación.
- La capa de **Presentación**: engloba las aplicaciones consumidoras de información obtenida de la base de datos mediante protocolos de Internet / Intranet como http y estándares de intercambio de información como XML. Además proporciona la interfaz necesaria para presentar información y reunir datos, para lo que se usan las XSL y los DTD. También aseguran los servicios de negocio necesarios para ofrecer las capacidades de transacciones requeridas e integrar al usuario con la aplicación para ejecutar un proceso de negocios. El cliente proporciona el contexto de presentación, generalmente un *browser* como Microsoft Internet Explorer, que permite ver los datos remotos a través de una capa de presentación HTML, JSP o Java Servlets. La capa de servicios de presentación es responsable de:
 - Obtener información del usuario.
 - Enviar la información del usuario a los servicios de negocios para su procesamiento.
 - Recibir los resultados del procesamiento de los servicios de negocios.
 - Presentar estos resultados al usuario.

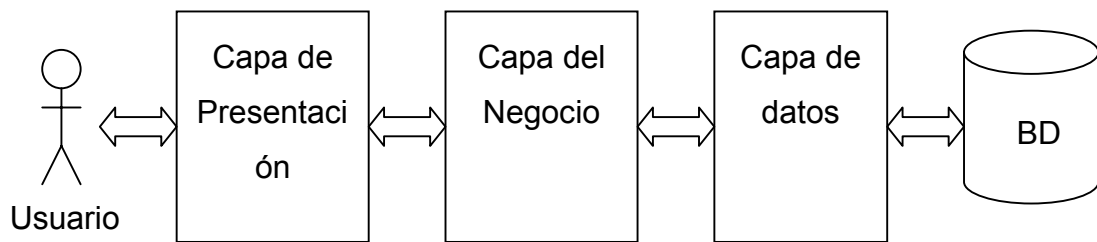


Figura 1

Como tecnología de desarrollo se ha elegido el enfoque de Orientación a Objetos y como arquitectura de sistemas se ha escogido la opción de Internet / Intranet. Ambas opciones se complementan con la elección de la arquitectura modelo en tres capas anteriormente mencionada y que hacen que se potencien las ventajas de cada una de ellas por separado.

En concreto, la elección de las opciones anteriores permite que el sistema sea mantenible, extensible y distribuible.

11.1. ARQUITECTURA FÍSICA DEL SISTEMA

La implantación del sistema se traduce en una estructura como la mostrada en la Figura 2. El sistema estará disponible en cada uno de los puestos de usuario (clientes) con conexión a Internet.

Además de los puestos cliente, se encontrará el servidor de las tres capas: Presentación, Datos y Aplicación.

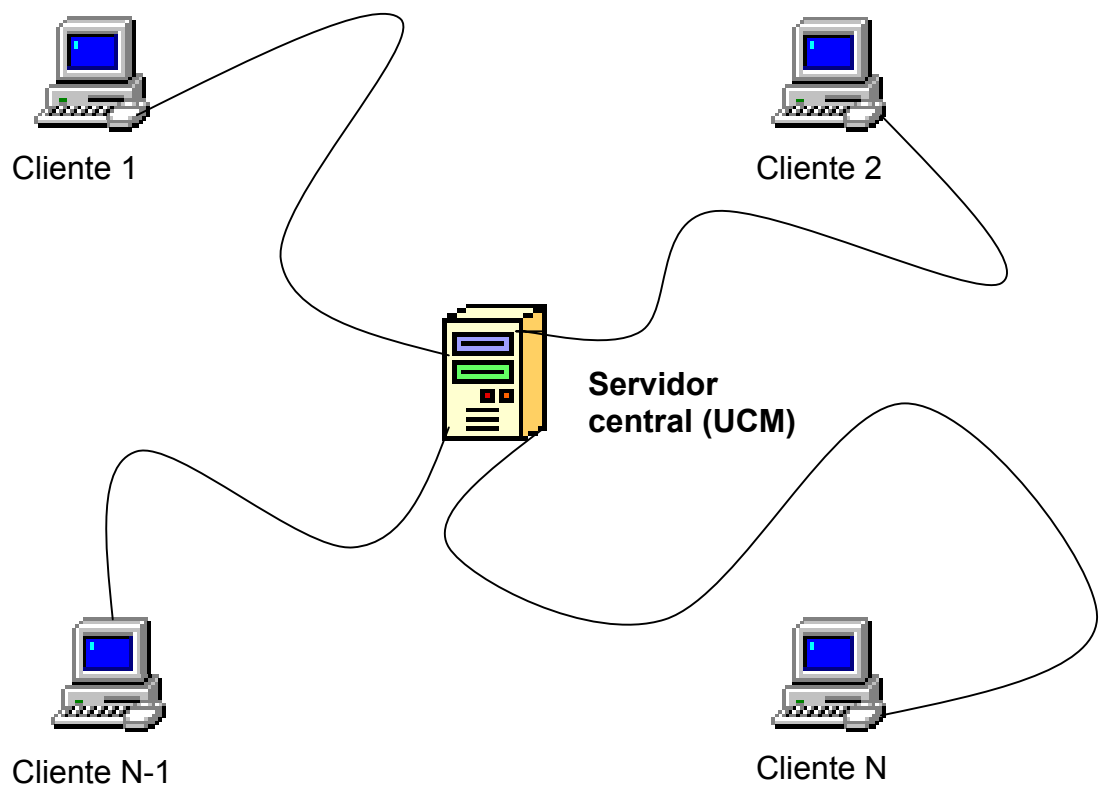
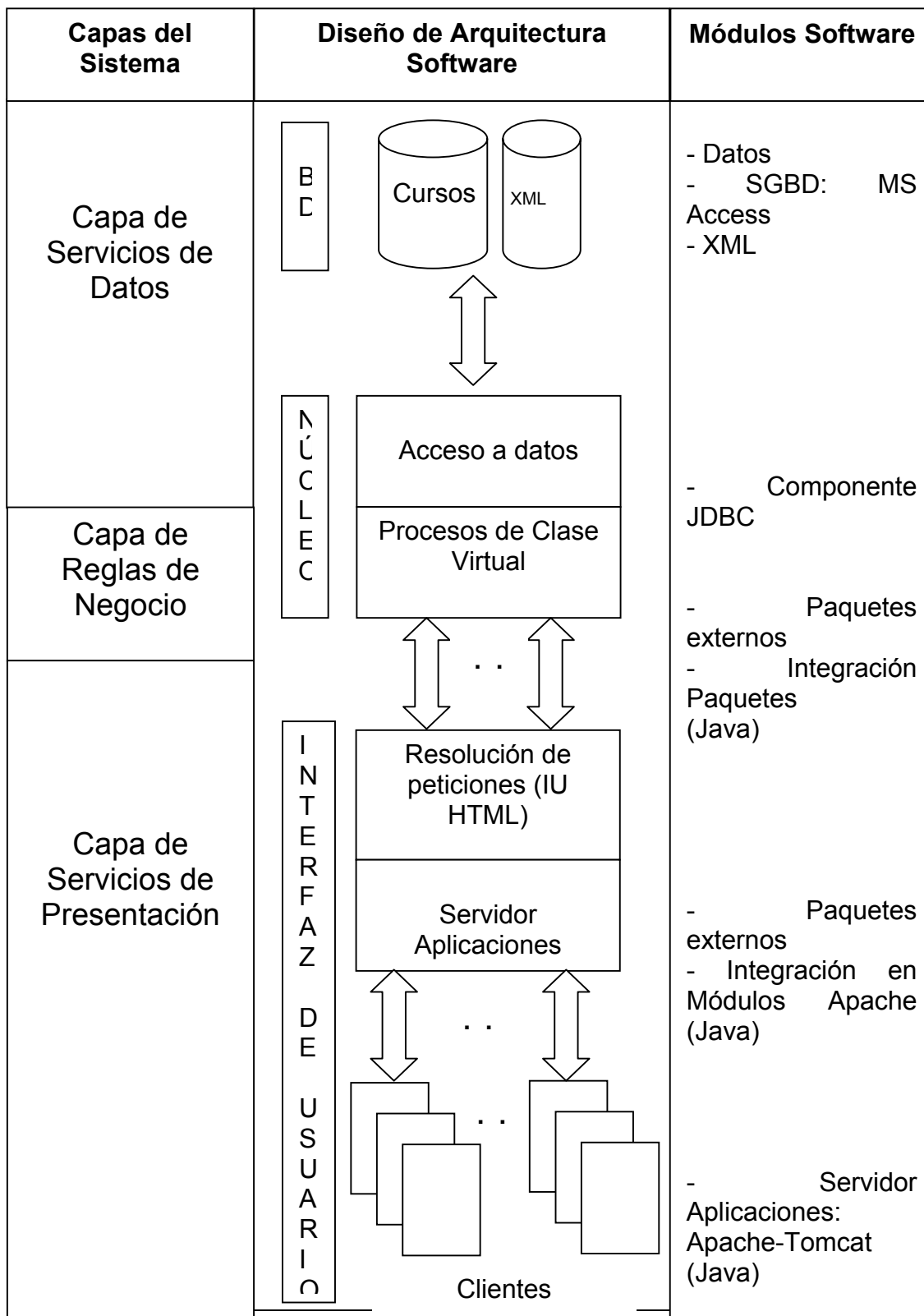


Figura 2. Arquitectura física del sistema.



11.2. ARQUITECTURA LÓGICA DEL SISTEMA

La arquitectura software del sistema se detalla en la figura que se muestra a continuación:





La columna central de la figura anterior muestra un diagrama de bloques del software del sistema, así como los flujos de información (datos, llamadas a funciones y envío de comandos) entre dichos bloques.

En la columna de la derecha se detallan dichos componentes software indicando el tipo de los mismos (lenguaje de desarrollo, nombre o tipo de programa). En la columna de la izquierda se muestra la correspondencia entre la división en tres capas mencionada inicialmente y los bloques de software utilizados.

Este apartado se limita a la descripción de los módulos software y a su relación, pero no da indicaciones de la localización física de los mismos, así como de la plataforma concreta que los ejecutará. Ambas informaciones están contenidas en los apartados *Arquitectura Física del Sistema* y *Plataforma*.

Los bloques de software representados son los siguientes:

Interfaz de Usuario: Según la figura anterior, los usuarios del sistema acceden al mismo por la parte inferior del cuadro, a través del interfaz de usuario. Para ello, los puestos cliente deben disponer de un navegador mediante el cual accederán a las páginas HTML de interfaz del sistema. Esta condición se cumple en cualquier máquina con sistema operativo Windows 95 o posterior y con Internet Explorer. En estos puestos cliente no se procesará ninguna parte de la aplicación, actuando como un puro visualizador de interfaz.

Estableciendo el paralelo con la arquitectura tricapa, la de Servicios de usuario es implementada por el software descrito en este apartado de Interfaz de Usuario.

Núcleo: En este bloque radica la lógica de la aplicación de Clase Virtual. Gestiona las aplicaciones que están en curso en cada momento y ejecuta los comandos en los que se traducen las funciones a las que tiene acceso el usuario. Como en muchas de las aplicaciones orientadas a datos, dichas funciones consisten en consultas a la base de datos y adecuación de los resultados con mayor o menor grado de procesamiento.

En la figura se han incluido en el mismo bloque de software, (núcleo), servicios de dos capas distintas: acceso a datos y reglas de negocio. Esto sería así sólo a nivel de programa (estaría en el mismo ejecutable), pero no a nivel de diseño ya que los servicios se hallan separados en diferentes paquetes, de tal manera que se da la separación lógica de ambas capas.

Los servicios de la Capa de Negocio acceden a la capa de datos utilizando JDBC y distribuye la información a la capa de presentación mediante un interfaz HTTP. El componente JDBC de acceso a datos transforma a consultas SQL los comandos de manejo de datos procedentes del usuario y las envía al gestor de base de datos para que las ejecute.



Base de Datos: Se ha incluido dentro de este apartado tanto el gestor de base de datos (Microsoft Access) como a la base de datos en sí misma. El problema de acceso a la información de la base de datos está resuelto por medio de una conexión ODBC a una base de datos Microsoft Access.

La solución diseñada refleja claramente la independencia entre los módulos software Interfaz de Usuario / Aplicación / Datos. La ventaja más evidente de esta solución es la obtención de un sistema fácilmente mantenible y compuesto por capas de software que pueden ser reutilizables desde otras aplicaciones o plataformas y de la misma manera, permitiría minimizar el coste de posibles modificaciones de funcionalidad, cambio del gestor de base de datos, adaptación a nuevas versiones del mismo,...

11.3. PLATAFORMA

11.3.1. Entorno de explotación

Tal como se mostraba en el apartado *Arquitectura física del sistema*, desde el punto de vista de la arquitectura física del sistema éste estará formado los siguientes elementos interconectados mediante Internet:

Servidor de datos, de aplicaciones y de interfaz de Usuario, que contiene y gestiona:

- La base de datos del sistema, proporcionando y actualizando los datos manejados por la aplicación. En él residen las funciones de salvaguarda y recuperación de datos, gestión de accesos concurrentes, multiusuario, administración de la base de datos, etc. Además de la información relativa al seguimiento de los alumnos en el curso almacenada en la base de datos MS Access "clasevirtual.mdb", tenemos información relativa a los alumnos almacenada en los XML que tienen como nombre el *login* de los estudiantes. De modo, que tanto la base de datos como los archivos XML son unidades de almacenamiento dentro del sistema.
 - Software: Sistema gestor de base de datos Microsoft Access y archivos XML.
- Aplicaciones: Implementa la lógica de la Clase Virtual y gestiona los servicios que presta respecto al mismo a los distintos usuarios (clientes) del sistema. Asimismo controla el acceso de los posibles clientes al sistema tanto desde un punto de vista lógico como físico. A través de la capa de acceso a los datos, envía las órdenes necesarias al servidor de datos, para el acceso a los mismos, (gestión de sesiones, consulta, alta, baja, modificación).
 - Software: Núcleo de paquetes integrados (Procesos y acceso a datos).



- Interfaz del usuario, que se encarga de la gestión y de las peticiones realizadas por los clientes de la aplicación, bien para transmitirlos a la capa siguiente, bien para formatear a HTML y reenviar los datos ya devueltos por dicha capa como resultado de una petición anterior. Es el responsable de la conexión de los clientes a la red así como de proporcionarles los recursos software necesarios para ejecutarla. El servidor de aplicaciones Apache-Tomcat permite una integración eficiente, gestión de contenidos, control de seguridad, y presentación de contenidos existentes en base datos por medio de programación en Java (Servlets/JSP). El servidor de aplicaciones será el encargado de servir las páginas WEB y realizar las peticiones de consultas contra la base de datos. Para tal efecto será necesaria la creación de páginas JSP (Java Server Pages) y/o Java Servlets que ataquen directamente el servidor de aplicaciones. A todos los efectos funciona como filtro con la base de datos. Para la capa de presentación tendremos también archivos XML que almacenan información de los cursos y las hojas de estilo XSL que sirven para dar formato y presentar dicha información.
 - Software: Servidor Apache y módulos de integración para Apache que generan páginas HTML (se ejecutan en los clientes).
 - Sistema operativo: Windows NT, 2000.
 - Hardware: Servidor de la UCM.

Clientes de la aplicación: Soportan el interfaz de usuario proporcionado por el servidor de Interfaz de Usuario mediante un navegador. Éste es el único software que debe contener la máquina de un cliente para permitir la explotación del sistema. Otro requisito será su alta como usuario en el sistema (Clase Virtual).

- Software: Navegador WEB, preferiblemente Internet Explorer.
- Sistemas operativos: Windows 9X, Windows NT/2000, o superiores.
- Hardware: PC de trabajo cada usuario del sistema cuyos requisitos hardware mínimos están fijados por el sistema operativo que use.

11.3.2. Entorno de desarrollo

- Servidor de datos, aplicaciones e Interfaz de usuario.
- Software: Sistema gestor de base de datos MS Access.
- Servidor Tomcat-Apache.
- Paquetes externos: JDBC, Tomcat, XML, JSP y Java Servlets.
- Entorno JBuilder 4.0.
- Sistemas operativos Windows 2000, Windows XP.
- Cliente Navegador Internet Explorer versiones 5.03, 6.0.
- Hardware: PC y portátiles de gama media – alta, Pentium III, IV.



11.4. DISEÑO DE LA BASE DE DATOS

Se usa para almacenar la información relativa al seguimiento del alumno durante el curso. Está implementada en MS ACCESS 2000 y se llama 'clasevirtual.mdb', siendo también 'clasevirtual' el DSN por el que se accede desde los JSP's y Servlets del proyecto.

11.4.1. Estructura

Usamos cuatro tablas para almacenar los datos:

- Tabla '**tiempo**':
 - Clave Primaria:

alumno	(texto 50)
curso	(texto 50)
pagina	(texto 50)
 - Campos:

tiempo	(Numérico)
ultima	(texto 50)
 - El significado es el siguiente :

El alumno 'alumno' ha estado viendo la página 'pagina' del curso 'curso' durante 'tiempo' segundos.

El campo 'ultima' indica si la página 'pagina' ha sido la última visitada para el alumno 'alumno' en el curso 'curso', y el valor de este campo puede ser SI/NO. Si este campo tiene el valor 'SI', significa que ha sido la última pagina vista, y en caso contrario tiene el valor 'NO'. Para un cierto alumno y curso sólo existe una página que tiene 'SI' como valor del campo 'ultima', y el resto de páginas para ese alumno y curso tienen el campo 'ultima' con el valor 'NO'.



- Tabla **'ejercicios'**:

- Clave Primaria:

alumno	(texto 50)
curso	(texto 50)
ejercicio	(texto 50)
nota	(Numérico)
tiempo	(Numérico)
fecha	(Fecha/Hora. El formato es de 'Fecha corta', esto es, dia / mes / año, o lo que es lo mismo, dd/mm/aa)

- El significado es el siguiente:

El alumno 'alumno' le ha dedicado 'tiempo' segundos al ejercicio 'ejercicio' del curso 'curso', siendo realizado dicho ejercicio en la fecha 'fecha' y obteniendo la nota 'nota'.

- Por ser clave primaria todos los campos, un alumno puede repetir todas las veces que quiera un ejercicio.

- Tabla **'exámenes'**:

- Clave Primaria:

alumno	(texto 50)
curso	(texto 50)
examen	(texto 50)

- Campos:

nota	(Numérico)
tiempo	(Numérico)
fecha	(Fecha/Hora. El formato es de 'Fecha corta', esto es, dia/mes/año, o lo que es lo mismo, dd/mm/aa)

- El significado es el siguiente:

El alumno 'alumno' le ha dedicado 'tiempo' segundos al examen 'examen' del curso 'curso', siendo realizado dicho examen en la fecha 'fecha' y obteniendo la nota 'nota'.

- Por como está puesta la clave primaria, un alumno no puede repetir varias veces el mismo examen.



- Tabla '**curso**':

- Clave Primaria:

- alumno (texto 50)

- curso (texto 50)

- Campos:

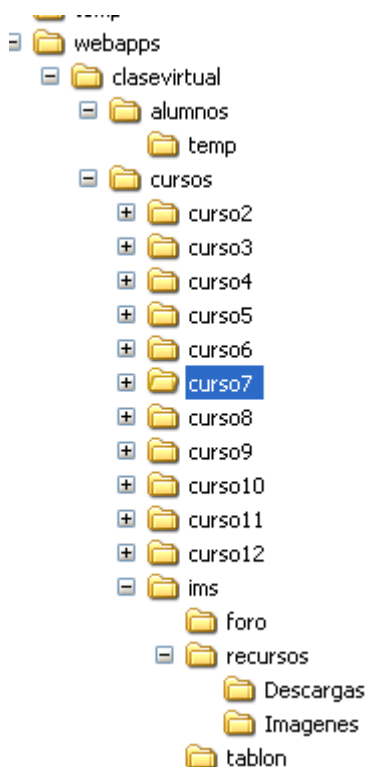
- visitas (Numérico)

- fecha (Fecha/Hora. El formato es de 'Fecha corta', esto es, dia/mes/año, o lo que es lo mismo, dd/mm/aa)

- El significado es el siguiente :

- El alumno 'alumno' ha visitado el curso 'curso' un número de veces igual a 'visitas', y la fecha de la última visita es igual a 'fecha'.

11.4.2. ORGANIZACIÓN DEL ARBOL DE DIRECTORIOS



En el directorio principal de la aplicación se encuentran los ficheros html y jsp usados en ella, además del fichero *cursoejemplo.zip*. También está el archivo *funcionesJs.js*, que almacena funciones de JavaScript usadas en algunos componentes.



11.4.2.1. Alumnos

En esta carpeta están los ficheros XML correspondientes a la información de los alumnos. El nombre de cada fichero es <usuario>.xml, donde <usuario> es el nombre de usuario del alumno. Además están las DTD's que validan estos XML.

Bajo esta carpeta también hay un directorio *temp*, usado para subir el xml del cliente cuando se importa un alumno.

11.4.2.2. Cursos

En esta carpeta se encuentran todas las DTD's que intervienen en los elementos que componen los cursos, los manifiestos que definen cada curso, las páginas que lo componen, los exámenes, el foro y el tablón. Asimismo están los XSL utilizados para la presentación de estos elementos.

También hay una carpeta por cada curso existente. Estas carpetas tienen como nombre el identificador del curso, y contiene a su vez tres carpetas (en esta carpeta no hay ningún fichero):

11.4.2.2.1. Foro

Contiene el fichero XML que almacena el foro de preguntas del curso (foro.xml).

11.4.2.2.2. Tablon

Contiene el fichero XML que almacena el tablón de anuncios del curso (tablon.xml).

11.4.2.2.3. Recursos

Esta carpeta almacena el contenido del curso. Corresponde al paquete IMS, y, por tanto, contiene el fichero imsmanifest.xml que define el curso, y todos los recursos ficheros XML correspondientes a dicho curso. También está el JSP *pagina.jsp*, que se utiliza para devolver el HTML generado para la presentación de las páginas. Dentro de esta carpeta se han incluido otras dos:

Descargas

En la que se guardan todos los ficheros usados desde los enlaces en el curso, para ser descargados por los usuarios, o para ser visualizados en una nueva ventana.

Imágenes

Donde se guardan las imágenes que se usan en el curso.

11.4.2.3. Imágenes

Las imágenes utilizadas en la aplicación.

11.4.2.4. Profesores

Ficheros XML de definición de cada profesor (<nombre_usuario>.xml), y la DTD que define la estructura de estos ficheros.



11.4.2.5. Universidad

La DTD universidad.dtd, que define la estructura de dos ficheros: universidad.xml, que almacena información acerca de los cursos que existen en la <e-aula>, y cursosprov.xml, que almacena esa información (en realidad sólo un subconjunto de la información utilizada por un curso en funcionamiento) para los cursos provisionales.

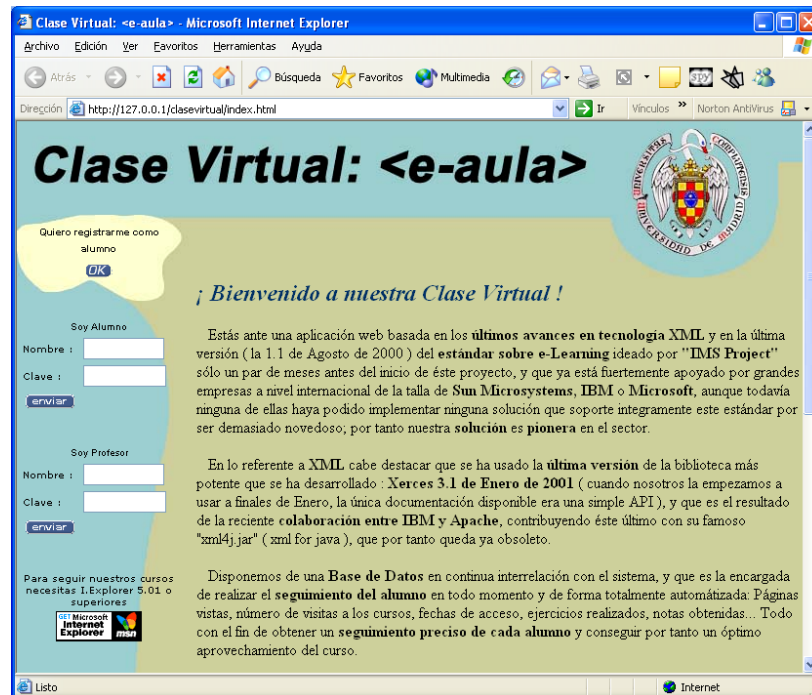
11.4.2.6. WEB-INF/classes

Este directorio, cuyo nombre viene dado por el funcionamiento del servidor TOMCAT, almacena las clases compiladas correspondientes al código fuente.

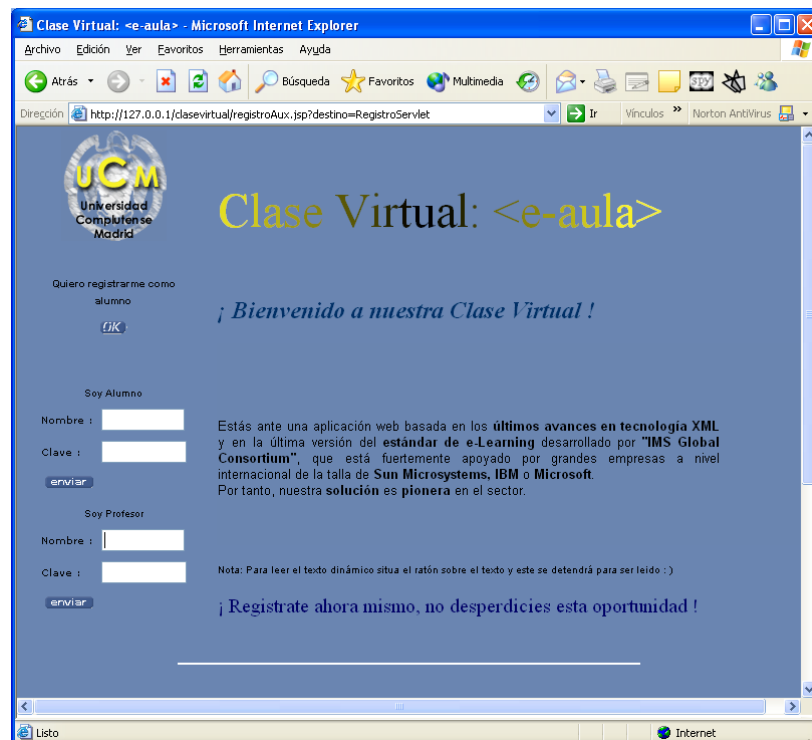


12. APÉNDICE A: COMPARATIVA DE INTERFACES

La página principal de la versión anterior es:

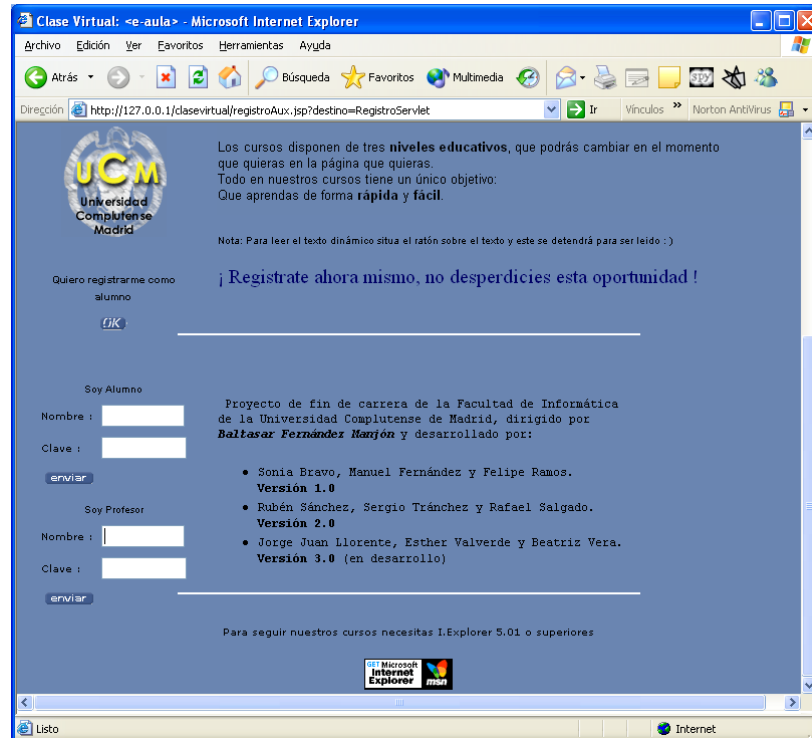


La página principal actual es la siguiente:

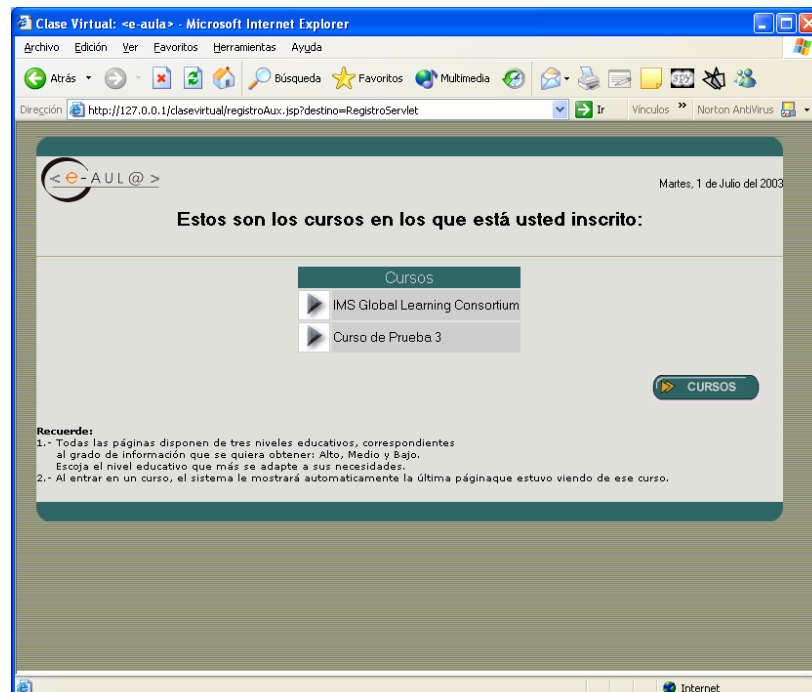




Aunque nos desplacemos por la página principal hacia abajo, el menú flotante nos dará la opción de registrarnos en la Web constantemente:

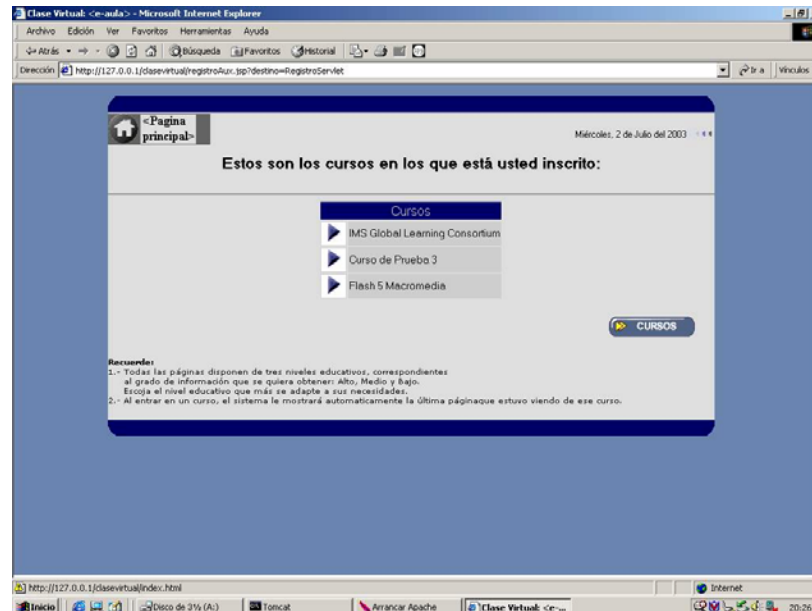


La página para la elección de los cursos antigua es la siguiente:

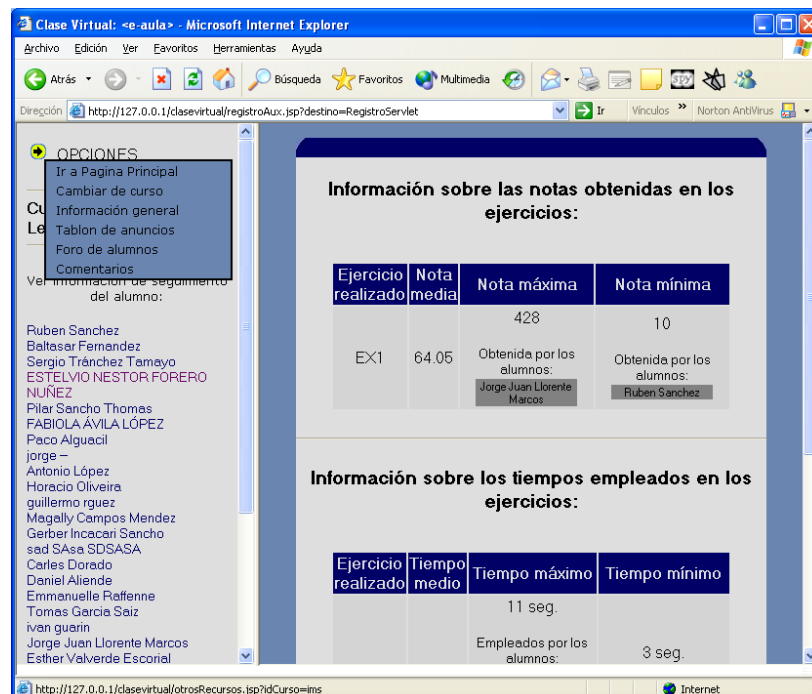


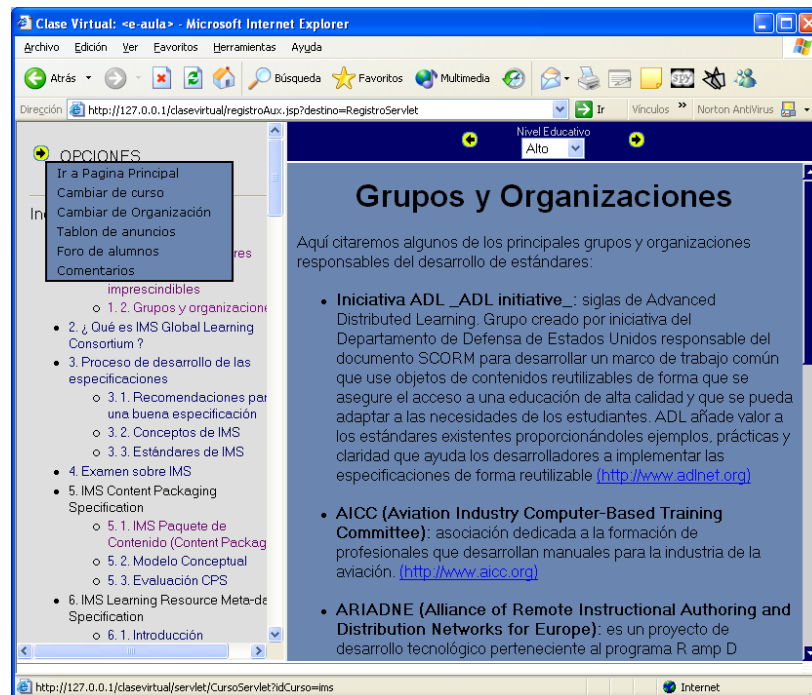


A continuación se muestra la pantalla actual en la que el logo de <e-aul@> cambia al pasar el ratón convirtiéndose en un enlace a la página principal.

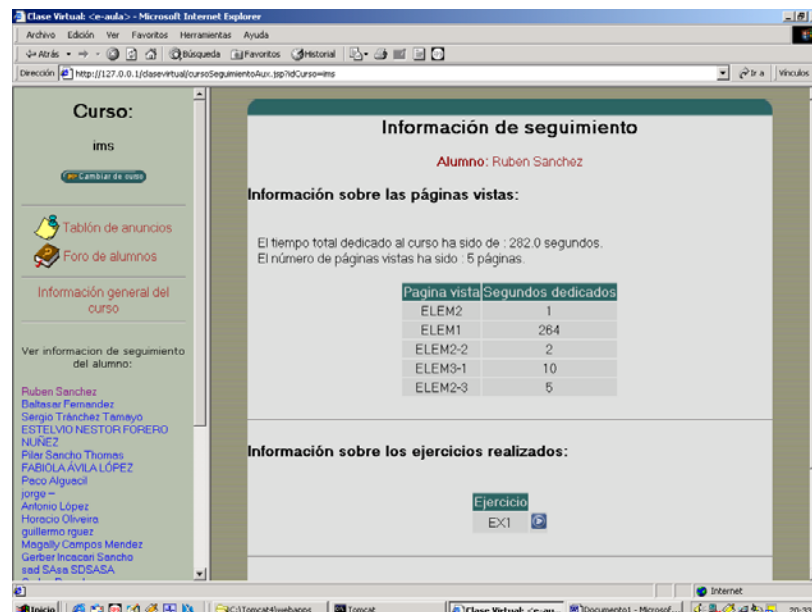


Las dos siguientes pantallas corresponden a la versión actual y muestran la mejora de la caja de links dinámicos.





En la siguiente imagen se ve la interfaz antigua en donde los botones ocupan gran parte de la pantalla.





13. APÉNDICE B: IMPLEMENTACIÓN

13.1. CAMBIOS EN EL FICHERO PAGINA2.DTD

Se ha modificado el elemento **párrafo** para reflejar los cambios realizados en el formato de texto. Los elementos nuevos se explican a continuación:

```
<!ELEMENT párrafo ((texto | fotosola | flash | foto | enlace | lista | finlinea | negrita | cursiva | subrayado | tamtexto | colortexto)+)>
<!ATTLIST párrafo
    nivel CDATA #REQUIRED
>
```

Se ha añadido el elemento **lista** para permitir incluir viñetas para clasificar el texto. La diferencia entre estas listas y el elemento *listaparrafos* que ya se encontraba en la versión anterior es que este último solo permite crear una enumeración con viñetas simples de párrafos (separados por una línea en blanco) frente al elemento introducido, que permite además listas ordenadas (a través del atributo, que dice si está ordenada o no) y no existe separación mediante línea en blanco entre sus componentes.

Los elementos que forman la lista son: otra *lista* (para permitir anidamiento de listas, estén ordenadas o no) y el elemento *LI*.

```
<!ELEMENT lista ((LI | lista)+)>
<!ATTLIST lista
    ord (no | yes) #REQUIRED
>
```

El siguiente elemento es necesario para insertar viñetas o números en una *lista*.

```
<!ELEMENT LI ((texto | fotosola | flash | foto | enlace | lista | finlinea | negrita | cursiva | subrayado | tamtexto | colortexto)+)>
```

Los siguientes elementos se han añadido para permitir los formatos **negrita**, **cursiva** y **subrayado**.

```
<!ELEMENT negrita (#PCDATA)>
<!ELEMENT cursiva (#PCDATA)>
<!ELEMENT subrayado (#PCDATA)>
```

El elemento que viene a continuación, **tamtexto**, permite introducir texto de tamaño distinto al predeterminado. Hay que tener en cuenta que en el atributo *tamaño* se introduce un número mayor o menor que el actual que es 3.

```
<!ELEMENT tamtexto (#PCDATA)>
<!ATTLIST tamtexto
    tamaño CDATA #REQUIRED
>
```

Para introducir texto de otro color, se introduce un elemento **colortexto** y el color (en inglés) en el atributo.



```
<!ELEMENT colortexto (#PCDATA)>
<!ATTLIST colortexto
    color CDATA #REQUIRED
>
```

También es posible crear enlaces a páginas Web o ficheros mediante el elemento, **enlace**. El enlace al recurso se introduce en el atributo.

```
<!ELEMENT enlace (#PCDATA)>
<!ATTLIST enlace
    recurso CDATA #REQUIRED
>
```

Para separar varios trozos de texto (plano o formateado) con un carácter fin de línea, está el siguiente elemento, **finlinea**. Así, se permite introducir un trozo de texto en negrita, cursiva,... dentro de texto plano en la misma línea.

```
<!ELEMENT finlinea (#PCDATA)>
```

13.2. CAMBIOS EL LOS FICHEROS MOSTRAR PAGINA *.XSL

En estos ficheros se han modificado las plantillas correspondientes a los elementos mencionados en el apartado anterior, para que se presenten en la página Web en la forma deseada. Los cambios realizados son los siguientes:

```
<xsl:template match="texto">
    <xsl:value-of/>
</xsl:template>
<xsl:template match="negrita">
    <b>
        <xsl:value-of/>
    </b>
</xsl:template>
<xsl:template match="cursiva">
    <i>
        <xsl:value-of/>
    </i>
</xsl:template>
<xsl:template match="subrayado">
    <u>
        <xsl:value-of/>
    </u>
</xsl:template>
<xsl:template match="tamtexto">
    <font>
        <xsl:attribute name="size">
            <xsl:value-of select="@tamaño"/></xsl:value-of>
        </xsl:attribute>
        <xsl:value-of/>
    </font>
</xsl:template>
<xsl:template match="colortexto">
    <font>
        <xsl:attribute name="color">
            <xsl:value-of select="@color"/></xsl:value-of>
        </xsl:attribute>
        <xsl:value-of/>
    </font>
</xsl:template>
```




```
</font>
</xsl:template>
<xsl:template match="lista">
  <xsl:choose>
    <xsl:when test=".[@ord='yes']">
      <OL>
        <xsl:apply-templates/>
      </OL>
    </xsl:when>
    <xsl:otherwise>
      <ul type="square">
        <xsl:apply-templates/>
      </ul>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
<xsl:template match="LI">
  <li>
    <xsl:apply-templates/>
  </li>
</xsl:template>
<xsl:template match="enlace">
  <A>
    <xsl:attribute name="HREF">
      javascript:nt=window.open('<xsl:value-of
select="@recurso"/>');nt.focus();
    </xsl:attribute>
    <xsl:value-of/>
  </A>
</xsl:template>
<xsl:template match="finlinea">
  <div></div>
</xsl:template>
```



14. APÉNDICE C: CURSO IMS

14.1. INTRODUCCIÓN A LOS ESTÁNDARES

Este texto se basa en el documento 'Getting Started with eLearning Standards' y en el documento 'Making Sense of Learning Specifications and Standards: A Decision Maker's Guide to their Adoption' publicado por The MASIE Center, un centro dedicado a investigar como se pueden usar las nuevas tecnologías como Internet para aplicarlas a la formación o educación virtual, para lo cual ofrece una serie de servicios (como laboratorios, seminarios...). Este centro tiene muy en cuenta los avances y opiniones de la asociación 'e-Learning CONSORTIUM' formada por corporaciones, agencias gubernamentales, proveedores de servicios de aprendizaje virtual, etc. que aprenden y comparten sus experiencias en común para avanzar lo más posible en el e-Learning (el término e-Learning hace referencia a la formación o el aprendizaje que está preparado, extendido y gestionado usando gran variedad de tecnologías para la educación, lo que hace que pueda ser desplegado local o globalmente).

Estándares de Aprendizaje (Learning Standards)

Han surgido debido a la creciente demanda de cursos virtuales que hay en la actualidad, a los que se les exige que sean reutilizables y portables (puedan ser llevados a diferentes plataformas sin mayores problemas). Esto junto con la necesidad, por parte de las organizaciones, de buscar la mejor manera de ofrecer a los alumnos de estos cursos una información y unas actividades o tareas específicas en cada caso, ha hecho que se de cada vez más importancia a cómo y dónde se guarda la información en el sistema, cómo recuperarla y como revisarla lo más rápidamente posible... El uso de estándares de aprendizaje para los cursos basados en las nuevas tecnologías reporta las siguientes ventajas:

- Mezcla y encaja contenidos provenientes de muy diferentes sitios.
- Desarrolla contenido intercambiable que puede ser reutilizado, mezclado o separado muy fácilmente.
- Asegura a los compradores que no se van a ver atrapados por una única empresa propietaria de la tecnología que compren.
- Asegura que las inversiones tecnológicas en este campo sean prudentes y con poco riesgo.
- Incrementa la efectividad del aprendizaje permitiendo una mayor personalización de los cursos para cada estudiante.
- Mejora la eficiencia para el desarrollo y control del contenido de los cursos.
- Mejora la eficiencia para el desarrollo y control del contenido de los cursos.

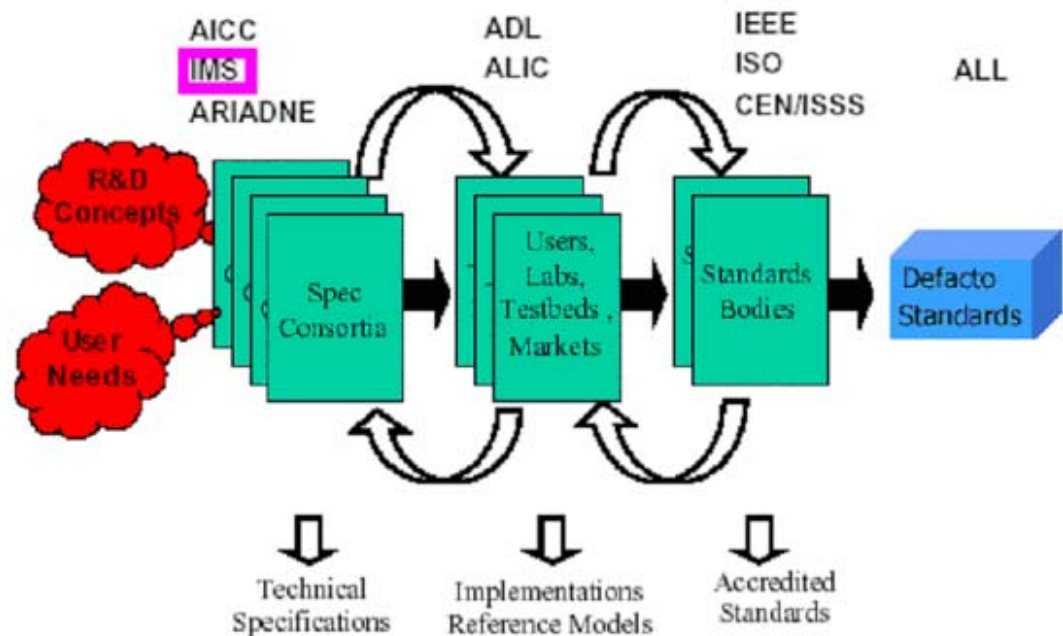
Estas ventajas se pueden resumir en las siguientes cinco palabras: interoperabilidad (el sistema puede ser usado con otros sistemas), reutilización (trozos del sistema (objetos de aprendizaje) puedan ser reutilizados), manejabilidad (el sistema puede seguir la pista de determinada información sobre los alumnos y los contenidos), accesibilidad (el alumno puede acceder a los contenidos que necesite en cada momento), durabilidad (siguiendo los estándares se evita que el sistema se quede obsoleto).

14.1.1. Términos imprescindibles

En relación con los estándares son imprescindibles los siguientes términos:

- **Especificación:** Es una descripción documentada, que si se certifica debidamente por una organización acreditada puede llegar a ser un estándar. Para ello pasa antes por diferentes fases y desarrollos.
- **Estándar:** que puede ser de dos tipos:
 - " de jure Estándar ": cuando es dado por alguna corporación acreditada como ISO que certifica su validez.
 - " de facto Estándar ": cuando es adoptado por la gran mayoría aunque no haya sido acreditado por nadie.

Un modelo de la evolución de los Estándares se ve en la siguiente figura:



Primero se investiga para identificar posibles soluciones a las necesidades de los usuarios (R amp D), cuando aparece una solución buena, ésta se documenta mediante una detallada especificación por escrito (Specification Development), esto es lo que ocurre con el consorcio IMS que dedica grupos de gente a puntualizar la



documentación de estas especificaciones para que puedan ser implementadas. Luego, estas especificaciones se ponen en manos de los usuarios que dicen lo que opinan de ellas (Testing) y así llegaran a alcanzar el estado de Estándar Internacional Acreditado cuando ya sea una especificación completa y robusta, revisada por una organización acreditada.

Cada organización que aparece en el gráfico tiene un papel distinto en el modelo por lo que no compiten entre ellas.

- **Objeto de aprendizaje " Learning Object " (LO):** es un trozo de contenido del curso con un determinado objetivo para el aprendizaje. El contenido del curso se divide en trozos que puedan ser usados por separado o añadidos dinámicamente unos a otros, para permitir así a los alumnos el seleccionar la información más relevante para ellos en cada momento. Antes el aprendizaje se organizaba en lecciones y cursos que tenían un objetivo, ahora esto se ha dividido todavía más en piezas de contenido con instrucciones que, como se ha mencionado antes, pueden ser unidas dinámicamente con otras para dar la información exacta en el momento justo a un determinado estudiante. Definir y entender estos objetos de aprendizaje ha sido un desafío porque necesitan ser vistos dentro de un modelo de contenidos conceptual y global (Conceptual content Model).
- **SCORM:** es un modelo de referencia para el desarrollo de contenido de aprendizaje y su distribución, que sirve para comprobar la efectividad de una colección de especificaciones individuales y estándares. El departamento de Defensa de Estados Unidos se quiso asegurar de que, independientemente de la fuente o programa base, cualquier rama de su departamento pudiera usar, intercambiar, controlar, seguir y rehusar sus tecnologías, contenidos y datos de aprendizaje y esta documentación pasó a llamarse the Sharable Content Object Reference Model (SCORM). SCORM trabaja con organizaciones como IMS, IEEE o AICC para integrar sus especificaciones en un modelo que sea usable y cohesionado y define las relaciones entre los diferentes estándares.
- **LMS:** siglas de " Learning Management System ", software que automatiza la administración de cursos. Para ello sigue los pasos de los cursos y los alumnos registrados y da la información a los que tienen que controlarla.
- **No confundir " conformance " (adaptado o adaptabilidad) con " compliance "(complacencia):** El primero se define como comprobar si una implementación o producto sigue los requisitos impuestos por un estándar o una especificación y si sigue los requisitos necesarios se certificará ese producto (tener en cuenta que un producto o estándar puede ser bueno sin estar certificado y que el que un producto siga un estándar no garantiza que será mejor ni de mejor calidad), y el segundo,



que hay que procurar no usar en este campo, significa complacencia con los estándares.

- **Meta-data (datos que describen otros datos):** necesario por el constante crecimiento del e-Learning que dificulta el encontrar la información que es realmente relevante. Esto es de lo que se encargan los Meta-datos " Meta-data ", de facilitar el proceso de identificación y descripción del contenido para el aprendizaje de tal manera que se pueda encontrar, juntar y dar en contenido exacto a una determinada persona. Los Meta-datos se pueden aplicar a cualquier tamaño y tipo de contenido para el aprendizaje, lo que permite que todos los niveles de contenido puedan ser encontrados y rehusados. Así será muy fácil encontrar un trozo de texto o una ilustración, una página de un libro, un capítulo de un curso o incluso el curso completo. Pero también se pueden aplicar a las personas, lugares o cosas, por lo que podremos encontrar nombres, direcciones, habilidades_ PODEMOS ENCONTRAR O SELECCIONAR LA INFORMACIÓN NECESARIA EN CADA MOMENTO Y OFRECERSELA A LA PERSONA O RECURSO QUE LA NECESITA. Los Meta-datos tienen cuatro usos principales:
 - Categorización: cuando se usan para añadir valor mediante la organización de la información en categorías. Un ejemplo lo tenemos en las categorías de búsqueda que tiene el buscador Yahoo. Para que esto tenga utilidad cuando buscamos información en diferentes sistemas, países, organizaciones_ es necesario que se use en todos ellos el mismo estándar de Meta-datos.
 - Taxonomías (niveles jerárquicos que pueden ser asociados a contenido de aprendizaje): es más eficiente organizar el contenido en taxonomías relacionadas entre sí que en categorías. Las taxonomías de Meta-datos permiten a los diferentes sistemas y estructuras el ser reconocidos, trasladados y entendidos.
 - Reutilización: Al estar organizada la información, pasa a ser más reutilizable. Lo cual tiene la ventaja de que se crea una vez y se puede reutilizar varias veces.
 - Reunión o montaje dinámico: la información puede ser unida o montada dinámicamente según se vaya necesitando y así se puede ofrecer justo lo que se necesita, cuando se necesita.
- El estándar de Meta-Datos más extendido y aceptado es el " IEEE Learning Object Meta-Data " (LOM), que es incluido en las especificaciones de IMS y ADL. Si tu contenido soporta IMS y ADL, entonces seguro que se soporta este estándar implícitamente.



14.1.2. Grupos y Organizaciones

Aquí citaremos algunos de los principales grupos y organizaciones responsables del desarrollo de estándares:

- **Iniciativa ADL _ADL initiative_:** siglas de Advanced Distributed Learning. Grupo creado por iniciativa del Departamento de Defensa de Estados Unidos responsable del documento SCORM para desarrollar un marco de trabajo común que use objetos de contenidos reutilizables de forma que se asegure el acceso a una educación de alta calidad y que se pueda adaptar a las necesidades de los estudiantes. ADL añade valor a los estándares existentes proporcionándoles ejemplos, prácticas y claridad que ayuda los desarrolladores a implementar las especificaciones de forma reutilizable (<http://www.adlnet.org>)
- **AICC (Aviation Industry Computer-Based Training Committee):** asociación dedicada a la formación de profesionales que desarrollan manuales para la industria de la aviación. (<http://www.aicc.org>)
- **ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe):** es un proyecto de desarrollo tecnológico perteneciente al programa R and D costeado por la Unión de Europea. (<http://ariadne.unil.ch>)
- **IEEE (Institute of Electrical and Electronics Engineers):** es un comité internacional que trabaja en el desarrollo de estándares técnicos, prácticas recomendadas y guías para sistemas eléctricos, electrónicos, de ordenadores o de comunicaciones. Su especificación más conocida es " Learning Object Meta-data " LOM, que define un estándar para el uso de los Meta-Datos. Tanto las organizaciones IMS como ADL (comentadas después) usan elementos de LOM en sus especificaciones (<http://ltsc.ieee.org>)
- **IMS Global Learning Consortium (Instructional Management System):** es un consorcio global con miembros de organizaciones educativas, comerciales y gubernamentales dedicado a definir y distribuir especificaciones basadas en XML y libres para productos e-Learning. IMS tiene dos objetivos claros:
 - Definir especificaciones técnicas para la interoperabilidad (habilidad de coger componentes desarrollados en una zona con una serie de herramientas o plataformas y usarlas en otro sitio con diferentes herramientas o plataformas)de las aplicaciones y servicios en el aprendizaje distribuido, y
 - Respalda la incorporación de las especificaciones IMS en los productos y servicios a nivel mundial. Lo que permitiría a



diferentes autores de todo el mundo trabajar juntos gracias al uso de ambientes de trabajo comunes. (<http://www.imsproject.org>)

(<http://www.adlnet.org>)

Las especificaciones IMS más conocidas son: IMS Meta-Data, IMS Content Packaging y IMS QTI.

- **ISO (International Organization for Standardization):** federación mundial no gubernamental de organizaciones de estándares de 140 países, uno de cada país. Promueve el desarrollo de estándares y toda clase de actividades relacionadas con facilitar el intercambio de servicios y productos. (<http://www.iso.org>)

14.2. ¿ QUÉ ES IMS GLOBAL LEARNING CONSORTIUM ?

'IMS' significa 'Instructional Management Systems' que se podría traducir por algo así como 'Sistemas de gestión de la enseñanza', y en un principio éste era su cometido, pero en nuestros días IMS se está centrando en estándares para servidores de conocimiento, para contenidos dedicados al aprendizaje, y en la introducción de estos estándares en la empresa. Es por todo esto que IMS es solamente un nombre, y no hay que fijarse en lo que significan sus siglas.

IMS es un intento por conseguir una especificación para el desarrollo del potencial de Internet como entorno de formación, permitiendo que contenidos y entornos de aprendizaje de múltiples autores puedan trabajar juntos.



Logo IMS Global Learning Consortium

*En su dirección Web <http://www.imsproject.org>
podrás encontrar todo tipo de información sobre
sus actividades.*

IMS reúne un conjunto de organizaciones académicas, comerciales y gubernamentales que trabajan en construir la arquitectura de Internet para el aprendizaje, entre las que están importantísimas empresas del mundo de la informática: Microsoft, Sun Microsystems, Cisco, Apple, Oracle, IBM

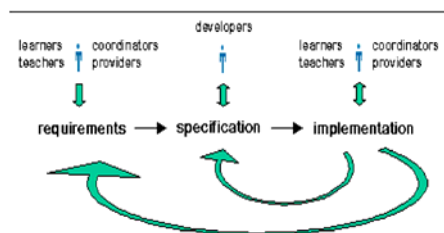
No cabe duda de la importancia del trabajo que este grupo viene desarrollando, ya que la adopción de sus especificaciones como un estándar internacional supondrá el impulso definitivo para los Entornos Virtuales de Enseñanza: el Aprendizaje a través de Internet.

14.3. PROCESO DE DESARROLLO DE LAS ESPECIFICACIONES

IMS Project enfatiza, en prácticamente todos sus documentos oficiales, que su especificación sólo define formatos y protocolos para intercambios de información que ocurren dentro y con otros entornos IMS, pero no pretende prescribir a los desarrolladores como deben hacer sus aplicaciones educativas ni pretende crear una metodología instruccional.

Para el desarrollo de estas especificaciones, se asumió como punto de partida unos requerimientos generados por los usuarios (profesores, estudiantes, proveedores, coordinadores) y desarrolladores de recursos de aprendizaje por Internet. Y es a partir de ahí desde dónde se inicia la elaboración de las especificaciones.

Posteriormente estas especificaciones sufrirán adaptaciones producidas por la propia implementación de los cursos, la cual producirá nuevos requerimientos que influirán de nuevo en dichas especificaciones, de manera directa (IMS se da cuenta de sus limitaciones) o indirecta (son los usuarios y desarrolladores los que informan a IMS), cerrándose así el ciclo evolutivo.



Ciclo del proceso generador de especificaciones

14.3.1. Recomendaciones para una buena especificación

1. **Los contenidos necesitan ser modularizados** el sistema debe ser escalable: Este principio es una consecuencia lógica de entender el aprendizaje como un conjunto de intercambios entre partes. Supone que los programas de aprendizaje deben ser pensados como una colección de unidades de aprendizaje independientes permitiendo diferentes combinaciones para diferentes contextos y diferentes alumnos. Esta habilidad para modularizar los contenidos facilitará un incremento en la producción y calidad de los materiales.
2. **Deseo de interoperatividad:** Garantiza que una herramienta construida para trabajar en un ambiente de aprendizaje, lo hará también en otros.
3. **Deseo de Personalizar y Extender:** La especificación de IMS estará pensada para que los diseñadores la puedan adaptar a los requerimientos específicos de sus usuarios. Las especificaciones



ayudan a definir una base común, desde la cual diferentes comunidades de aprendizaje pueden desarrollar sus propios requerimientos y especificaciones. Respecto a la extensibilidad la especificación debe estar abierta a la incorporación de nuevos estamentos, etiquetas de meta-datos... en la medida en que el desarrollo y las necesidades de los usuarios así lo determine.

14.3.2. Conceptos de IMS

- **Objeto de aprendizaje (Learning Object):** Es una cantidad de información que tiene significado propio. No tiene un tamaño definido, es decir, puede ser un módulo educativo o un conjunto de varios módulos con un mismo objetivo, una lección, un ejercicio, un video, etc.
- **Contenedores:** Encapsulan una cantidad dada de contenido. Pueden ser un elemento primario (texto, vídeo, gráficos, herramientas) y/o otros contenedores. Para facilitar el almacenamiento, y búsqueda, cada contenedor tendrá un juego de campos de meta-datos.
- **Meta-datos:** Son datos acerca de los datos. Es información descriptiva acerca de recursos de aprendizaje.
- **Paquete:** Representa una unidad de contenido usable (y re-usable) que puede ser un curso completo en sí mismo o una colección de cursos. Un paquete debe ser independiente, es decir, debe contener toda la información necesaria para usar los contenidos para aprender.
- **Fichero para intercambiar paquetes (Cacahué Interchange File):** Es un fichero comprimido (preferiblemente en formato zip) que contiene un paquete.

14.3.3. Estándares de IMS

Todos los estándares que vienen a continuación pertenecen a IMS y están formados por tres documentos:

1. Information Model: Describe en que consiste el estándar.
2. XML Binding: Implementa el Information Model.
3. Best Practice Guide: Guía para un uso correcto de los estándares.

Estos estándares serán explicados con más detalle a lo largo del curso:

- **Content Packaging Specification v1.1.2:** Define un conjunto estándar de estructuras que pueden ser usadas para intercambiar contenido entre sistemas que quieren importar, exportar, agregar y disgregar paquetes de contenido.



- **Learning Resource Meta-Data Specification v1.2.1:** Describe los nombres, definiciones, organización y restricciones de los elementos Meta-datos de IMS. La especificación de Meta-Datos hace más eficiente el proceso de encontrar y usar un recurso dando una estructura de elementos definidos que describen o catalogan los recursos junto con los requisitos sobre cómo los elementos deben ser usados y representados.
- **Enterprise Specification v1.1:** Da soporte a la interacción entre los sistemas de gestión de información y los sistemas empresariales tales como gestión de recursos humanos, administración de estudiantes y sistemas de gestión de bibliotecas.
- **Question & Test Specification v1.2:** Proporciona soporte al intercambio de cuestionarios, tests, exámenes... entre diferentes autores, publicadores y otros desarrolladores de contenido.
- **Learner Information Packaging v1:** Permite guardar la información del estudiante e intercambiarla con distintos sistemas.
- **Learning Design Specification v1:** Es el estándar más reciente. Describe cómo diseñar el "entorno de aprendizaje". Permite transformar un paquete de contenido en una unidad de aprendizaje.
- **Digital Repositories Interoperability Specification v1:** Dar recomendaciones para la interoperabilidad de las funciones más comunes.
- **Reusable Definition of Competency or Educational Objective Specification v1:** Define un modelo de información para describir, referenciar e intercambiar definiciones de competencias, principalmente en el contexto del aprendizaje online y distribuido.
- **Simple Sequencing Specification:** Define de qué manera y en qué orden se deben presentar las diferentes actividades y contenidos a los alumnos.

14.4. VER EXAMEN EN INTERNET

14.5. IMS CONTENT PACKAGING SPECIFICATION

14.5.1. IMS Paquete de Contenido (Content Package)

El "IMS Content Packaging Information Model" describe estructuras de datos que se utilizan para proporcionar interoperatividad del contenido basado en Internet, con herramientas para la creación de dicho contenido, sistemas de gestión de información y entornos en tiempo de ejecución" (IMS Content Packaging Information v1.1.2, Introduction)

El objetivo de este estándar es definir y estandarizar un conjunto de estructuras que pueden usarse para intercambiar contenido. Estas estructuras dan la base para permitir a los desarrolladores e implementadores de software crear materiales instructivos que interoperan con herramientas de autorización, sistemas de gestión de información y entornos en tiempo de ejecución que han sido desarrollados independientemente por varios desarrolladores de software.

El alcance de esta especificación está orientado a definir la interoperabilidad entre diferentes sistemas que quieren importar, exportar, agregar y disgregar paquetes de contenido.

Las versiones futuras de este estándar dirigirán sus requisitos teniendo en cuenta los modelos de contenido de datos y la comunicación entre entornos en tiempo de ejecución y los sistemas de gestión de información.

14.5.2. Modelo Conceptual

El modelo conceptual del estándar IMS Content Packaging es el siguiente:

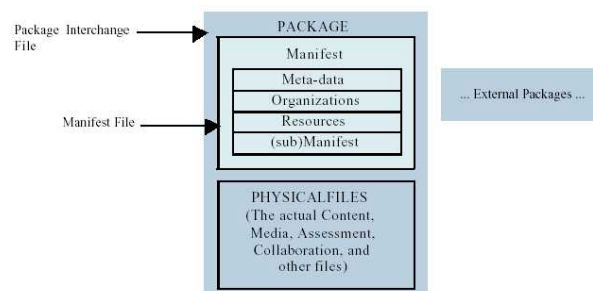


Figure 2.1 IMS Content Packaging scope.

Diagrama correspondiente a un paquete IMS

Un paquete IMS está formado por dos partes:

1. Un fichero XML que describe la organización y los recursos del paquete. A este documento se le llama manifiesto IMS del paquete.
2. Los ficheros físicos descritos por el manifiesto IMS.

Al fichero compuesto por estas dos partes (por ejemplo, comprimiéndolo en formato .zip, .jar, .cab) se le llama 'fichero de intercambio de paquete'. Se recomienda usar el formato .zip

El paquete es, por tanto, un directorio lógico que incluye el manifiesto IMS, cualquier documento de control XML que reference (como un fichero DTD o XSD) y sub-directorios que contienen los recursos físicos reales.



A continuación se explica cada de las partes, arriba mencionadas, que componen el paquete IMS:

1. El manifiesto es un fichero XML que describe parte de un curso, un curso completo o una colección de cursos. Es decir, describe al paquete en sí mismo. El nombre del fichero está predefinido y es **'imsmanifest.xml'** (imprescindible que esté en minúsculas). Si este fichero no existe, el paquete no es un paquete IMS y no puede ser procesado. También está predefinida su localización: el manifiesto y cualquiera de los ficheros XML que lo soportan deben estar situados en la raíz del paquete. Está compuesto por los siguientes elementos:
 - Meta-datos: Un elemento de XML que describe el manifiesto. Puede aparecer cero o una vez.
 - Organizaciones : Es un elemento XML que contiene cero, uno o más elementos "organización", los cuales describen las distintas organizaciones del contenido dentro de un manifiesto, es decir, las distintas formas de visualizar o presentar el contenido.
 - Recursos: Contiene cero o más elementos "recurso". Cada elemento recurso hace referencia a un recurso o elemento de comunicación real necesitado por el manifiesto, incluyendo meta-datos sobre esos recursos y referencias a ficheros externos. El elemento recursos debe contener todos los recursos referenciados en el manifiesto.
 - (Sub)Manifiestos: Permiten el anidamiento de manifiestos.
2. Los ficheros físicos podrán ser de cualquier tipo: son los elementos de comunicación reales, ficheros de texto, gráficos y otros recursos en sus directorios tal como están descritos por el manifiesto o sub-manifiesto.

Diagrama de los elementos de un manifiesto:

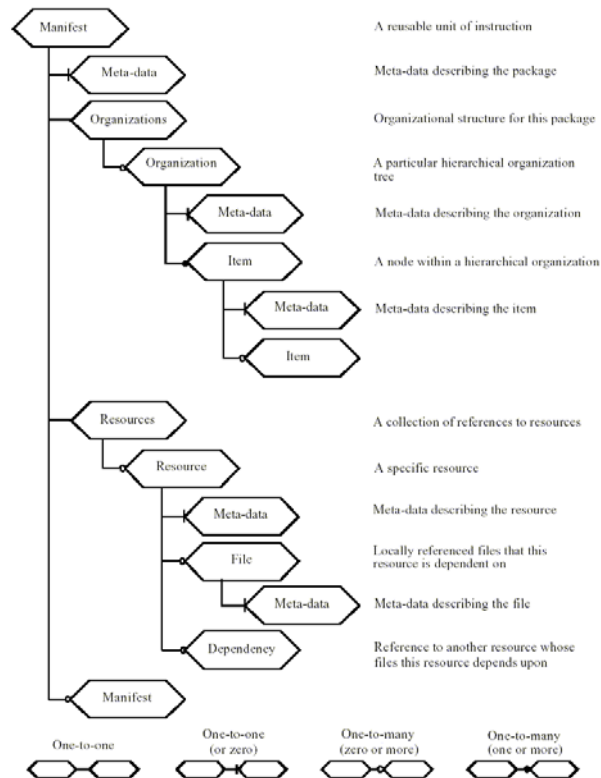


Figure 4.1 Manifest elements.

Elementos de un manifiesto

Aquí se pueden observar todos los elementos que componen un manifiesto y la interrelación que se da entre ellos.

Cuando dentro de un elemento "item" de una "organización" se hace referencia a un (sub)manifiesto, debe interpretarse como sigue:

- Si el (sub)manifiesto no incluye ningún elemento "organización", la referencia no puede resolverse y se trata como una referencia vacía (null identifierref).
- Si el (sub)manifiesto incluye un elemento "organización" puede ser usado en el contexto en que se usa el contenido y el nodo raíz de esta organización se unirá con el "item" que le referencia. Si en ambos se especifica el mismo atributo pero con valores distintos, el valor definido en el elemento "item" sobrescribirá al valor definido en la "organización".

El siguiente diagrama representa cómo el contenido de un (sub)manifiesto se une virtualmente con el contenido del manifiesto que le referencia. Los círculos representan a los items de una organización.

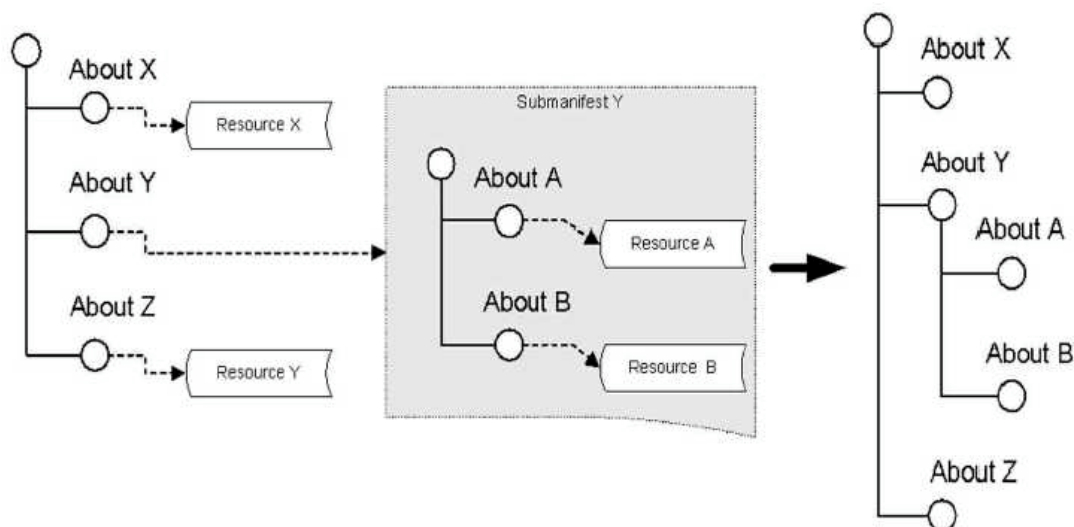


Figure 5.1 Merging <organization> from a (sub)Manifest.

Unión de la organización de un (sub)manifiesto con un ítem

El diagrama que viene a continuación representa cómo el contenido de un (sub)manifiesto se une virtualmente con el contenido de un ítem que tiene hijos propios y cómo los hijos del ítem tienen precedencia sobre los que se unen.

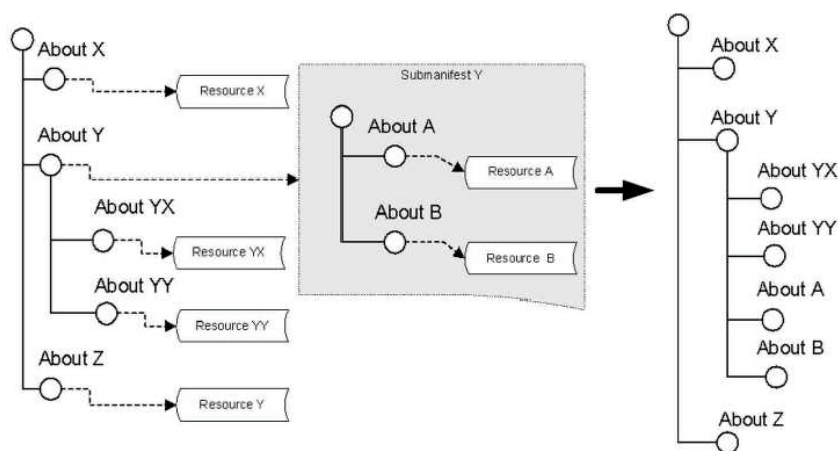


Figure 5.2 Merging <organization> from a (sub)Manifest when the referencing <item> has children.

Unión de un (sub)manifiesto con un ítem cuando el ítem que hace la referencia tiene hijos

[Para descargar los ficheros del estandar originales pincha aquí](#)



14.5.3. Ver evaluación en Internet

14.6. IMS LEARNING RESOURCE META-DATA SPECIFICATION

14.6.1. Introducción

Los desarrolladores y diseñadores de materiales de aprendizaje disponen de una gran variedad de herramientas software para crear los recursos de aprendizaje. Desafortunadamente, esta gran variedad de herramientas software de distintos vendedores, producen materiales de enseñanza que no comparten un mecanismo común para encontrar y usar estos recursos.

Pueden usarse etiquetas descriptivas para indexar los recursos de aprendizaje y hacer, por tanto, más fácil su búsqueda y uso. Tales etiquetas son 'datos sobre los datos', es decir, meta-datos (meta-data). Un ejemplo de meta-datos es la etiqueta de una lata de comida, que describe los ingredientes, peso, coste... de la lata. Otro ejemplo es la tarjeta de un libro en una biblioteca, la cual describe el libro, su autor, tema, localización dentro de la librería...

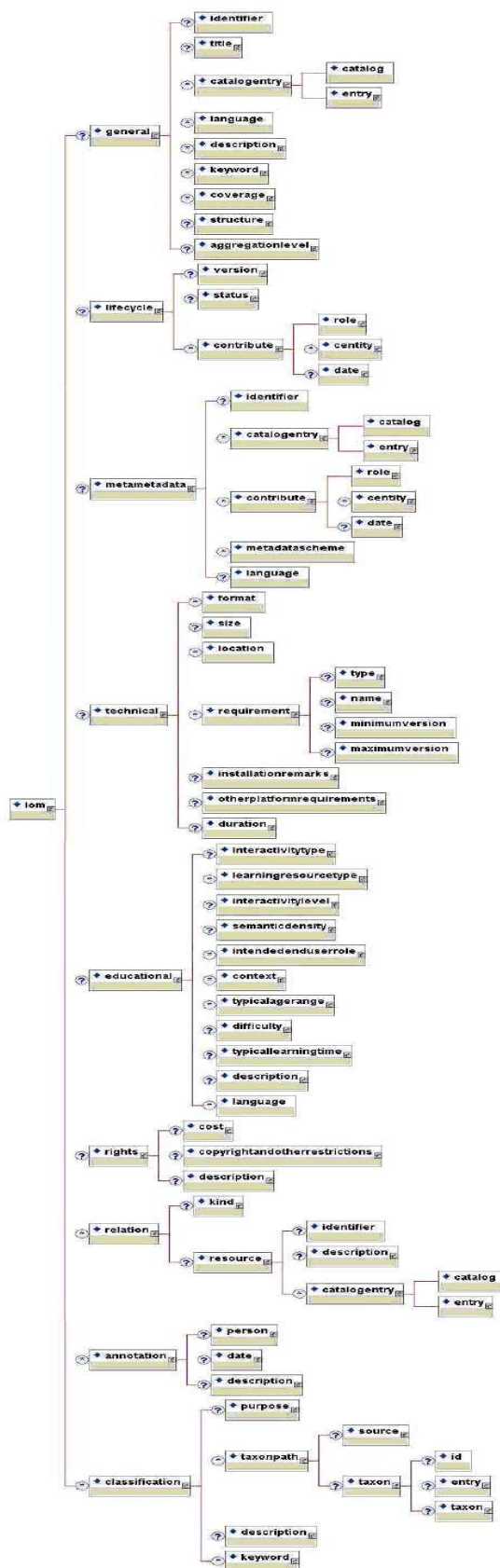
Por tanto, **una especificación de meta-datos hace que el proceso de búsqueda y uso de un recurso sea mucho más eficiente**, dando una estructura de elementos definidos que describa o catalogue el recurso de aprendizaje, junto a los requisitos sobre cómo usar los elementos y representarlos.

Los esfuerzos de IMS para desarrollar estándares abiertos, basados en el mercado para aprendizaje on-line (incluyendo especificaciones para meta-datos) se unieron en 1997 con los del NIST (National Institute for Standards and Technology) e IMS empezó a colaborar con el Proyecto ARIADNE (un proyecto europeo con un esfuerzo activo en definir meta-datos). En 1998 IMS y ARIADNE se unieron a IEEE, que formó las bases para el IEEE Learning Object Meta-Data (LOM) Draft Standard. **El IEEE LOM Draft Standard** define un conjunto de elementos de meta-datos que pueden usarse para describir procesos de aprendizaje. Esto incluye los nombres de los elementos, definiciones, tipos de datos y longitud de los campos. La especificación también define una estructura conceptual para los meta-datos. También intenta soportar una definición consistente de los elementos de meta-datos para múltiples implementaciones, pero no incluye información sobre cómo representar meta-datos en un formato que la máquina entienda.

14.6.2. Modelo Conceptual

El modelo conceptual para las definiciones de Meta Datos es una jerarquía en forma de árbol. Cada elemento en la jerarquía tiene una definición, tipo de dato y valores permitidos específicos.

El modelo conceptual del estándar IMS Learning Resource Meta-Data (elementos y estructura) es el siguiente:





El elemento raíz es obligatoriamente: **<lom>**. Únicamente debe aparecer una y sólo una vez en una instancia de Meta-Datos IMS XML.

Tiene un atributo: **xmlns** el cual indica el Namespace de los Meta-Datos IMS.

Los detalles sobre cada elemento individual se pueden encontrar en el documento IMS Learning Resource Meta-Data Information Model:

http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd_info1p2p1.html

En el documento IMS Learning Resource Meta-Data XML Binding se encuentra una descripción exhaustiva de los elementos con ejemplos de su uso en XML. Este documento se puede encontrar en la siguiente dirección Web:

http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd_bind1p2p1.html

A continuación se describen los elementos que contiene **<lom>**. Por ser demasiado extenso, no se describirán los sub-elementos; para mayor información consultar los citados documentos.

1. **<general>**: Información general que describe al objeto de aprendizaje como un todo. Este elemento aparece cero o una vez.
2. **<lifecycle>**: Relata la historia y el estado actual del objeto de aprendizaje y a aquellas personas u organizaciones que han afectado al objeto durante su evolución. Este elemento aparece cero o una vez.
3. **<metametadata>**: Agrupa información sobre la instancia de meta-datos en sí misma (en lugar del objeto de aprendizaje que esta instancia describe). Es decir, son "meta-datos sobre los meta-datos". Este elemento aparece cero o una vez.
4. **<technical>**: Reúne los requisitos y características técnicas del objeto de aprendizaje: formato, tamaño, localización, requisitos, etc. Este elemento aparece cero o una vez.
5. **<educational>**: Son las condiciones de uso del recurso: tipo de interactividad, densidad semántica, a quién va dirigido, entorno, dificultad y tiempo de duración de trabajo con el recurso. Este elemento aparece cero o una vez.
6. **<rights>**: Se reflejan las condiciones de uso del recurso y si tiene coste, copyright u otras restricciones. Este elemento aparece cero o una vez.
7. **<relation>**: Relaciona al recurso con otros objetos de aprendizaje, diciendo el tipo de relación existente entre el recurso que describimos y el objeto mencionado. Este elemento aparece cero o más veces.
8. **<annotation>**: Son comentarios hechos por personas sobre el uso educativo del objeto de aprendizaje. Puede incluir la persona, fecha y el comentario en sí. Este elemento aparece cero o más veces.



9. **<classification>**: Son descripciones de una característica del recurso por entradas en alguna clasificación. Se incluyen los nombres de las clasificaciones (cualquier taxonomía "oficial" reconocida) con la descripción y palabras clave relativas al propósito concreto tratado. Este elemento aparece cero o más veces.

En el siguiente enlace se muestra un fichero de ejemplo:

[Ejemplo de fichero de meta-datos](#)

Una instancia de meta-datos cumple con la especificación IMS Meta-Data si satisface los cuatro requisitos siguientes:

- La instancia debe contener uno o más elementos LOM (Learning Object Meta-Data).
- Todos los elementos LOM se usan tal y como están descritos por la especificación LOM.
- Los valores de los elementos están estructurados tal y como están definidos por la especificación LOM y esta información estructurada se lleva con la instancia. Esto significa que se debe mantener la agrupación en categorías y sub-elementos.
- Las ataduras deben llevar información equivalente sobre los meta-datos para que las conversiones entre ataduras no induzcan pérdidas de información tal y como está definido en la especificación.

Si la instancia contiene extensiones a la estructura LOM, entonces estas extensiones no reemplazan a ningún elemento de la estructura LOM.

Una aplicación cumple con la especificación IMS Meta-Data si satisface los dos requisitos siguientes:

- La aplicación debe ser capaz de procesar todos los elementos LOM.
- Si una aplicación recibe una instancia de meta-datos válida, la guarda y la transmite, entonces se conserva la instancia original durante la retransmisión.

Se puede encontrar una lista de taxonomías y vocabularios en el documento IMS Learning Resource Meta-Data Best Practice and Implementation Guide.

Los elementos que aparecen en esta lista pueden ser usados en campos concretos de uso, por esta razón no son "la práctica mejor" (best practice) sino una guía de práctica común que puede ayudar en un campo concreto. Este documento se puede encontrar en:

http://www.imsglobal.org/metadata/imsmdv1p2p1/imsmd_bestv1p2p1.html

[Para descargar los ficheros del estandar originales pincha aquí](#)



14.6.3. Ver evaluación en Internet

14.7. IMS ENTERPRISE SPECIFICATION

14.7.1. Introducción

El IMS Enterprise Information Model describe estructuras de datos que se utilizan para proporcionar la interoperatividad de sistemas de gestión de recursos de aprendizaje encaminados a Internet con otros sistemas de la empresa usados para dar soporte a las operaciones de la organización.

El objetivo del IMS Enterprise Information Model es definir un conjunto de estructuras que se puedan usar para intercambiar datos entre diferentes sistemas. Estas estructuras proporcionarán la base para la transmisión de información entre software desarrollado por distintos proveedores.

Los principales tipos de aplicaciones empresariales soportadas por este estándar son la administración de los modelos de enseñanza, la administración de los estudiantes, el mantenimiento de la 'biblioteca' de información, y los sistemas de recursos humanos.

Cabe destacar que el objetivo del IMS Enterprise Information Model está centrado en la definición de la interoperatividad entre sistemas que estén dentro de la misma empresa y no en resolver los problemas de integridad de datos, seguridad de información y otros problemas relativos al intercambio de datos.

14.7.2. Casos de uso

El alcance de la información del IMS Enterprise Specification esta enfocado a definir la interoperatividad entre sistemas que residen dentro de la misma empresa u organización

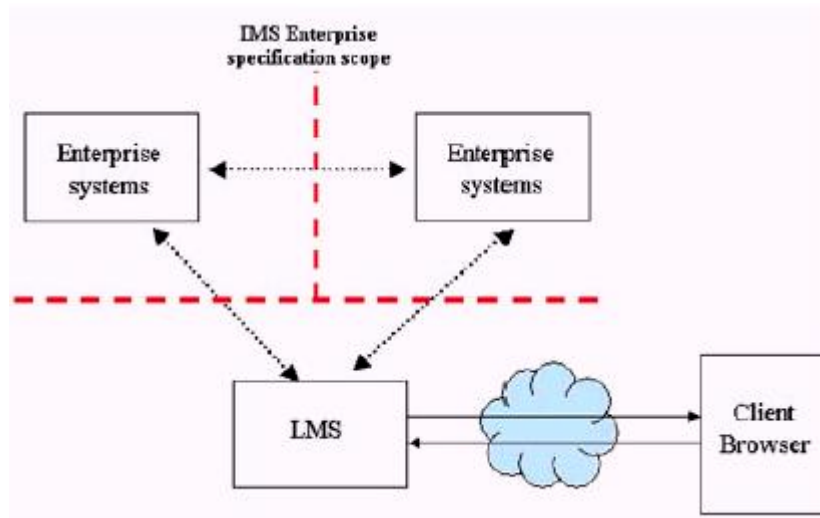
Los intercambios de datos pueden producirse entre empresas separadas, pero los documentos basados en el IMS Enterprise Specification no están dirigidos a resolver los asuntos de integridad de datos, comunicación, seguridad global... El IMS Enterprise Information Model está diseñado para soportar interoperatividad entre los siguientes cuatro componentes de proceso de negocio, los cuales requieren interacción entre LMS (Learning Management Systems) y estos tipos de empresa:

- **Mantenimiento de datos personales:** Típicamente los datos sobre las personas se gestionan en los sistemas de las empresas y se pasan al gestor de aprendizaje, cuando estos datos cambian en el sistema de la empresa, se necesita actualizar en el sistema de gestión de aprendizaje.

- **Gestión de grupos:** Los procesos de gestión de grupo pueden incluir datos de creación y planificación de clases y del continuo mantenimiento de los datos. un sistema fuente crea y mantiene grupos de información, los cuales necesitan ser compartidos con otros sistemas que están involucrados con funciones de gestión de grupos. El flujo de información de gestión de grupos no se produce necesariamente en un sentido; algunos datos deben ser actualizados por un sistema en blanco y pasados de nuevo al sistema principal.
- **Gestión de matriculas:** La gestión de matricula abarca la creación inicial de un miembro del grupo y cambios en los datos con el paso del tiempo. Ejemplos de gestión de matricula incluyen la matriculación en cursos y la asignación de instructores a los cursos.
- **Procesamiento del resultado final:** Se refiere a la evaluación y almacenamiento de los resultados del miembro del grupo (el grado final, los cursos completados...). Este proceso puede tener lugar en el LMSs o en el Enterprise system.

14.7.3. Arquitectura Básica

El modelo de arquitectura básica para Enterprise V1.1 Specification se muestra en la siguiente figura:

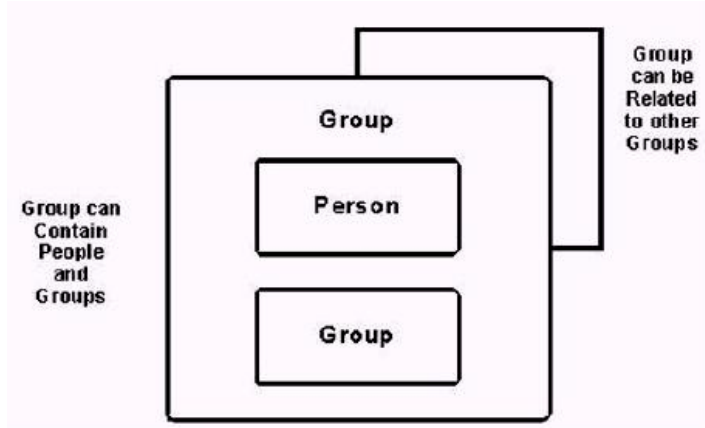


En esta arquitectura el alcance de IMS Enterprise Specification se muestra como una línea de puntos. El alcance de la interoperatividad es el modelo de datos de los objetos que están cambiando y no el modelo de comportamiento asociado o la infraestructura de comunicación requerida.

14.7.4. Modelo de Información Básico

Núcleo de los objetos de datos:

Una representación esquemática del cambio del núcleo de los objetos de datos usando IMS Enterprise Specification se muestra en la siguiente figura:



El núcleo de los objetos es:

- **Person:** Son los individuos que están emprendiendo algún tipo de estudio y/o actividad de grupo, por ejemplo son miembros de un curso particular.

La estructura personal solo contiene información usada para identificar el individuo y que es necesaria en los sistemas de aprendizaje. No está diseñado para ser un almacén para toda la información personal sobre un individuo.

- **Group:** Una colección de objetos relacionados con actividades de aprendizaje o individuos.

El grupo es un contenedor genérico usado para definir cualquier clase de actividades relacionadas, por ejemplo, un tutor de grupo, una clase, un currículo... no hay restricción en como las estructuras de grupo y subgrupo pueden ser usadas con respecto a contener otros grupos, personas...

- **Group Membership:** La estructura de miembros se usa para definir miembros de un grupo.

Un miembro puede ser una persona u otro grupo. Un grupo o una persona pueden ser miembros de cualquier número de grupos. El grupo y el miembro se identifican usando sus "sourceids".

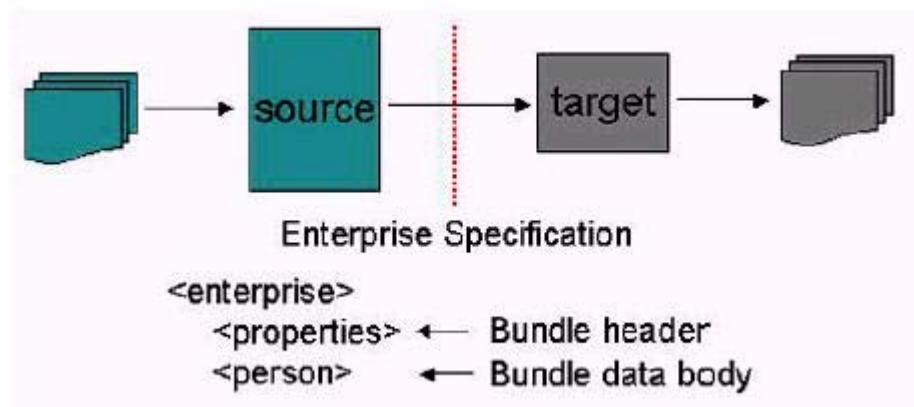
Una instancia de Empresa XML esta diseñada para contener cualquier numero estructuras de personas, grupos y miembros, pero el orden en el que tienen que aparecer está limitado a: todos los objetos de persona, seguido de todos los objetos de grupo, seguido de todos los objetos de miembro.

14.7.5. Modelo de Envío de Mensajes

IMS Enterprise Specification contiene un modelo de envío de mensajes muy simple, pensado para estar por debajo de los intercambios de datos entre los sistemas de comunicación de las empresas.

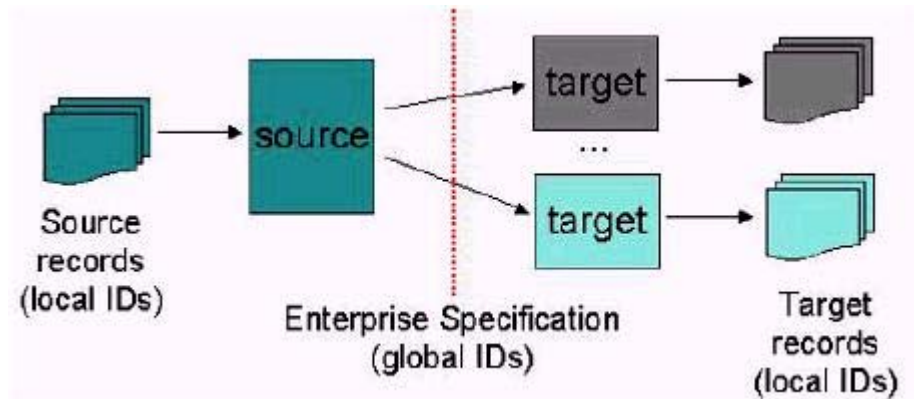
El mecanismo básico de intercambio de mensajes se muestra en la siguiente figura donde los sistemas de empresa "source" (fuente) y "target" (objetivo) intercambian una instancia de un fichero IMS Enterprise XML. La instancia XML consiste en un mensaje cabecera (llamado properties) y un mensaje de cuerpo (este contiene la mezcla adecuada de estructuras de persona, grupo y miembro de grupo).

Es importante recordar que la estructura de la instancia XML y el uso actual de XML no soporta como la misma información es contenida dentro de los sistemas de empresa fuente y objetivo.



Una suposición subyacente en un sistema basado en mensajes es que la secuencia de los mensajes se referirá a los mismos objetos en diferentes momentos. Por ejemplo: la creación, la actualización y la eliminación de un registro. Esto significa que los objetos deben tener un identificador único y que esa dirección/etiqueta es única entre los sistemas que se están comunicando. IMS Enterprise Specification denomina a estos identificadores "sourceid" del

objeto y consiste en una fuente (global y único en todos los sistemas de empresa) y un id (único dentro del sistema de empresa fuente).

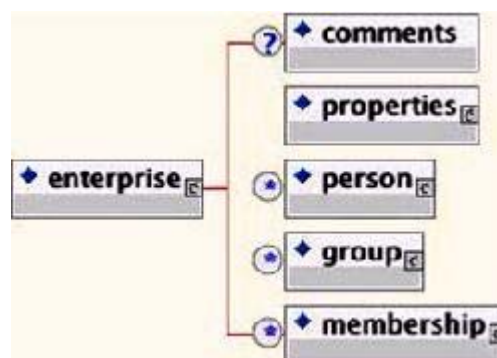


El uso del 'sourceid' para etiquetar los objetos que están siendo intercambiados da lugar al escenario mostrado en la figura 3.3 La consecuencia de esta aproximación es que cualquier objeto tendrá múltiples identificadores dependiendo de en que parte del sistema se esté considerando.

- El 'sourceid' es el identificador de intercambio y debe ser único para un sistema fuente particular.
- El identificador local dentro del sistema fuente. Debe ser un número de registro de base de datos, etc. El sistema fuente debe mantener una tabla de identificadores: cada identificador local con su correspondiente 'sourceid'.
- El identificador local dentro de los sistemas destino. En general el identificador local en cada sistema destino debe ser diferente. Es responsabilidad de los sistemas destino mantener la tabla local: para cada 'sourceid', su identificador local.

14.7.6. Descripción Conceptual de los Objetos de Datos

Elementos principales:





Extensiones Funcionales: extensiones que están incluidas para asegurarse que los usuarios de la especificación pueden añadir funcionalidad que es en otro caso excluido de la especificación (en las siguientes descripciones tabulares están denotadas por la estructura de datos 'extensión').

14.7.7. Estructura Tabular del Paquete Enterprise

Elementos Principales

Extensiones funcionales: extensiones que están incluidas para asegurarse que los usuarios de la especificación pueden añadir funcionalidad que en otro caso excluido de la especificación.

Descripción Tabular del Paquete Enterprise

Las tablas proporcionan una descripción conceptual de los elementos en los objetos de datos. Las columnas a las que se refieren son:

No: el número del elemento. Un elemento puede estar compuesto por sub-elementos. La numeración del esquema refleja estas relaciones

Name: el nombre descriptivo del elemento

Explanation: descripción del elemento.

Required: indica si el elemento es requerido:

- M= elemento obligatorio que debe ser incluido en el objeto de datos, si el elemento al nivel mas alto esta incluido
- C= elemento condicional. Existe o no dependiendo de los valores de otros elementos
- O= elemento opcional

Mult: multiplicidad del elemento

- Blank: instancia simple
- Number: máximo numero de veces que el elemento se repite
- N: sin limite, permitidas múltiples ocurrencias

Type: descripción de las reglas de formación del elemento.

- ID= elemento usado para identificar de manera única un objeto
- Code= valor del elemento que se obtiene de una lista de códigos
- Descripción= elemento descriptivo, lenguaje humano
- Flag= flag binario



- Enumerated= lista de opciones no numéricas predefinidas, por ejemplo: la lista definitiva de objetos.

En todos los campos se puede usar el conjunto de caracteres especificado por ISO 10646. El tipo puede incluir una descripción del conjunto de valores válidos para los sub-elementos.

Note: información descriptiva adicional sobre el elemento.

Tablas...

- [Enterprise Data Objects](#) describe los objetos de datos usados en la construcción del paquete IMS Enterprise
- ['person' Data Objects](#) describe los objetos de datos que son usados en la construcción de la información del sumario
- ['group' Data Objects](#) describe los objetos de datos que son usados en la construcción de los resultados detallados de evaluación
- ['membership' Data Objects](#) describe los objetos de datos que son usados en la construcción de la sección de resultados
- [Common Data Objects](#) describe los objetos de datos comunes que son usados para ayudar a otros objetos de datos
- ['extension' Definitions](#) lista el conjunto de nombres de extensión que son usados en el XML binding. Estos nombres son mapeados con el elemento 'extension' usado en las tablas 4.1 y 7.5 del modelo de información
- [Data Types](#) presenta las definiciones de los tipos de datos usados en la descripción tabular del modelo de datos de información

14.7.8. Conformidad

El propósito de esta declaración es proveer de un mecanismo a los clientes para comparar a los vendedores de sistemas y subsistemas.

No es obligatorio (para un vendedor) cumplir todos los rasgos de Enterprise Specification, pero debe detallar el nivel de cumplimiento con una 'declaración de conformidad'.

Esto está representado por:

- **Resumen de conformidad:** es un resumen que muestra, en términos coloquiales, las capacidades de una implementación particular con respecto a IMS Enterprise Specification

[Ver Tabla: Enterprise Conformance Summary](#)



- **Declaración de interoperatividad:** es una lista detallada que identifica todos los rasgos de las capacidades de la implementación en términos de las funciones IMS Enterprise Specification

Ver Tablas: Interoperability Statement

[Tabla Properties](#)

[Tabla Person](#)

[Tabla Group](#)

[Tabla Membership](#)

[Para descargar los ficheros del estandar originales pincha aquí](#)

14.7.9. Ver evaluación en Internet

14.8. IMS QUESTION & TEST INTEROPERABILITY SPECIFICATION

14.8.1. Introducción

El IMS Question & Test Interoperability Information Model Specification describe estructuras de datos que se utilizan para proporcionar interoperatividad entre sistemas de test y preguntas, particularmente aquellos que están basados en Internet.

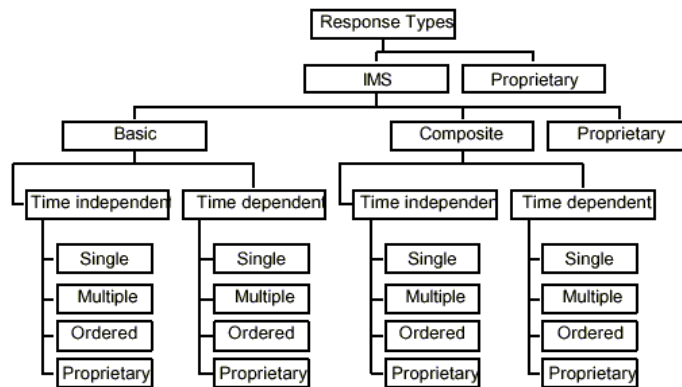
Existen tres partes básicas:

- **Assessment:** La unidad básica o test.
- **Sección:** Un contenedor de grupos de secciones e ítems que soportan un objetivo común.
- **Item:** El bloque más básico compuesto de pregunta y respuesta.

La principal ayuda que proporciona esta especificación es permitir a los usuarios la importación y exportación de sus preguntas (llamadas 'items' y agrupadas en secciones) y tests (llamados 'assesmentes' y que contienen secciones).

Todo esto requiere una especificación concisa y sin ambigüedades que cubra el más amplio rango posible de tipos de preguntas y tests. Esta especificación está construida de tal forma que es capaz de soportar tanto tests complejos como simples, y permitir extensiones del usuario que no comprometan la integridad del sistema.

Por ejemplo, lo siguiente es la taxonomía para los posibles tipos de respuesta



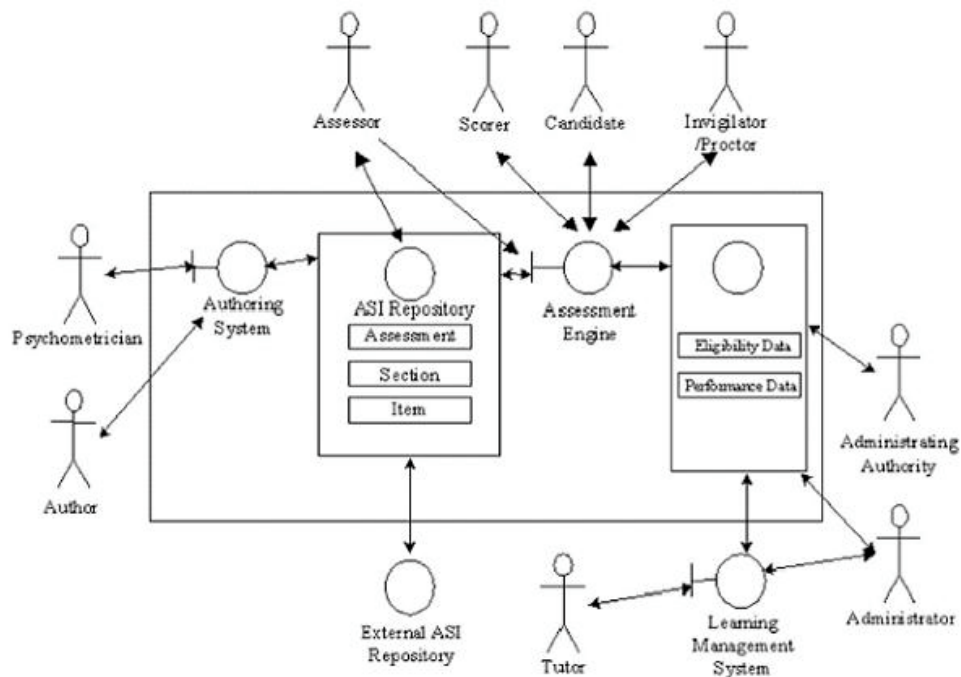
Taxonomía para una 'respuesta'

1. **Basic:** Sólo hay un tipo de respuesta.
2. **Composite:** Hay varios tipos de respuesta.
3. **Time Dependent:** Es importante el tiempo que se tarde en responder.
4. **Time Independent:** No es importante el tiempo que se tarde en responder.

[Para descargar los ficheros del estandar originales pincha aquí](#)

14.8.2. Casos de Uso

Los componentes de procesos (círculos) y las estructuras de datos (rectángulos) y los participantes (personas) de la especificación IMS QTI se muestran en la siguiente figura:



Los componentes clave del sistema de evaluación son:

- **Sistema creador:** el proceso que soporta la creación y edición de evaluaciones, secciones e ítems (ASIs).
- **Motor de evaluación:** el proceso que soporta la evaluación de las respuestas en términos de producción de puntuaciones relacionadas, evaluación y realimentación.
- **Sistema de manejo del aprendizaje:** el proceso/sistema responsable del manejo de la arquitectura de aprendizaje.
- **Depósito de datos de candidato:** la base de datos de la información específica del candidato.
- **Depósito de ASI:** la base de datos de los ASIs locales.
- **Depósito de ASI externo:** las bases de datos del ASI externo que serán importadas a través del uso de las especificaciones QTI.

Versión 1.2 IMS QTI: el modelo de información de ASI está relacionado con la especificación de las estructuras de datos de ASI. A lo largo de la descripción del modelo de información, el término 'view' se usará para describir la perspectiva del sistema como la percibe un participante. Distintos participantes tendrán acceso a funciones similares pero el contenido reflejará la naturaleza del actor.



Sólo 3 de los posibles casos de uso se presentan como ejemplos en esta versión de la especificación:

- **Autor:** creación y edición del ASIs;
- **Evaluación 'High-Stakes':** el examen de candidato;
- **Evaluación 'Low-Stakes':** apoyo del tutor que usa el ASIs;
- **Evaluación basada en el contenido:** contenido nativo interactivo basado sobre QTI-XML.

CASO DE USO AUTOR

La secuencia de procesamiento, respecto a las estructuras de datos ASI es:

- El autor lanza el sistema creador.
- Después crea o modifica, ítems, secciones y/o evaluaciones, que son exportadas usando esta especificación QTI y almacenadas en una base de datos externa.
- El autor puede importar ASIs que puede usar para crear nuevos ASIs.
- Una de las principales responsabilidades del autor, es determinar el tipo de respuesta. También es responsable de suministrar al actor información específica (puede ayudar a los usuarios a descubrir cómo usar el material).
- El psicólogo establece los pesos y los parámetros de los ítems en las evaluaciones.

CASOS DE USO DE LA EVACUACIÓN

'High-Stakes'

El motor de evaluación es responsable de realizar esta actividad. Es importante resaltar que la operación interna del motor de evaluación está fuera del alcance de esta especificación. Este caso de uso se ha incluido porque justifica algunos de los componentes estructurales que están definidos en el ASIs.

La secuencia es la siguiente:

- El asesor construye/selecciona los ASIs para usarlos a lo largo del procedimiento de evaluación. Estos ASIs se almacenarán en ciertas bases de datos internas.



- La evaluación se activa por el candidato, esta actividad está vigilada por el celador/censor. El candidato responde al ASIs y produce un conjunto de respuestas, que se almacenan de nuevo internamente. Las respuestas son un conjunto de identificadores de los ítems más la información asociada que caracteriza la respuesta.
- O sincrónicamente o asincrónicamente cada respuesta será evaluada por el procedimiento de respuesta para construir la puntuación inicial. Esta puntuación requiere el uso de un conjunto de reglas que se usan para definir parámetros llave a través de los cuales las preguntas serán evaluadas. La evaluación del ítem resultante se almacena en la estructura de datos Outcomes.
- El proceso de acumulación ahora toma parte en qué Outcomes se analizan y cotejan en términos de carga, etc. Definidos como parte de las estructuras de datos Section. Esta información se almacena como parte del registro Assessment.
- La fase final del proceso de evaluación es el Proceso de Evaluación Acumulada en el cual el registro Assessment además se procesa con respecto a las instrucciones de nivel de la estructura de datos Assessment.
- La fase final del proceso es la realimentación del registro Assessment a la selección de la actividad que puede resultar a su vez en una modificación del ASIs presentado al candidato.

'Low-Stakes'

El caso de uso del Tutor es similar al caso de uso Evaluación. Las diferencias son que el candidato recibirá un rango de información de realimentación incluyendo consejos y una o más soluciones posibles.

La secuencia es:

- El tutor construye/selecciona el ASIs que usará a través del procedimiento del curso. Este ASIs se almacenará en ciertas bases de datos internas. Los candidatos pueden poder actuar como su propio tutor con un cierto control sobre su selección de la actividad.
- El candidato activa la sesión del tutor. El candidato responde al ASIs y produce un conjunto de respuestas, almacenadas internamente de nuevo. Las respuestas son un conjunto de identificadores del ítem, incluyendo identificadores del tipo de respuesta más la información asociada que caracteriza la respuesta.



- Cada respuesta es evaluada por el proceso de respuesta para construir el ítem. Esta puntuación requiere el uso de un conjunto de reglas que se usan para definir parámetros llave a través de los cuales las preguntas serán evaluadas. La evaluación del ítem resultante se almacena en la estructura de datos Outcomes.

El resto del procesamiento es como en el caso de uso Assessment.

CASO DE USO BASADO EN EL CONTENIDO INTERACTIVO

Es posible usar el QTI-XML para realizar cualquier contenido de material de enseñanza. El contenido no tiene que usarse para cualquier forma de evaluación.

El proceso de desarrollo de es el siguiente:

- Una herramienta creadora se usa para construir el contenido y el esquema apropiado. Se debe usar una aplicación de ayuda siempre que se requiera una forma del contenido basada en preguntas, por ejemplo, una pregunta de elección múltiple. El autor debe asegurarse de que se crea el ítem completo, es decir, incluyendo el materia de presentación, el procedimiento de respuesta, realimentación y la descripción de los meta-datos. Después este material se exporta a su instancia QTI-XML equivalente.
- Para el intercambio de contexto interactivo entre un LMS y el motor representación del contenido se debe definir el correspondiente modelo de interacción. Los diferentes componentes del QTI-XML ahora son mapeados en su modelo de transacción. Esto puede resultar en la realimentación adecuada que está siendo enviada desde el LMS al sistema de representación del contenido usando de nuevo la representación QTI-XML.

14.8.3. Preguntas, Ítems y Respuestas

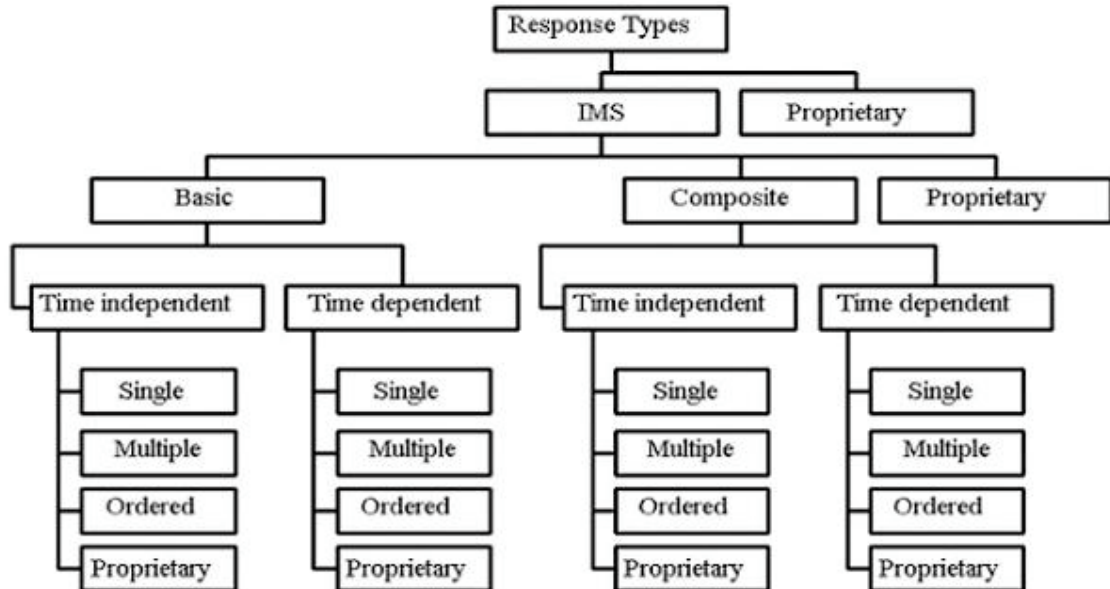
La terminología adoptada por el QTI es la siguiente: El ítem se define como el bloque fundamental que contiene una o más preguntas y respuestas. De esta forma, el concepto tipo de artículo, o tipo de pregunta, es inadecuado así como la identidad de referencia fundamental se basará en el tipo de respuesta. El tipo de respuesta es el identificador único para el tipo de la respuesta requerida.

Taxonomía del Tipo de Respuesta

La taxonomía de tipo de respuesta adoptada es mostrada en la siguiente figura. Un tipo de respuesta de IMS puede ser básico o compuesto (la tercera categoría es el grupo propietario):

- **Básico:** que contenga sólo un tipo sencillo de respuesta;

- **Compuesto:** un tipo de respuesta compuesto se refiere a una respuesta que actúa como un contenedor para la (sub-) respuesta, normalmente combinaciones diferentes de los tipos de respuesta básicos.

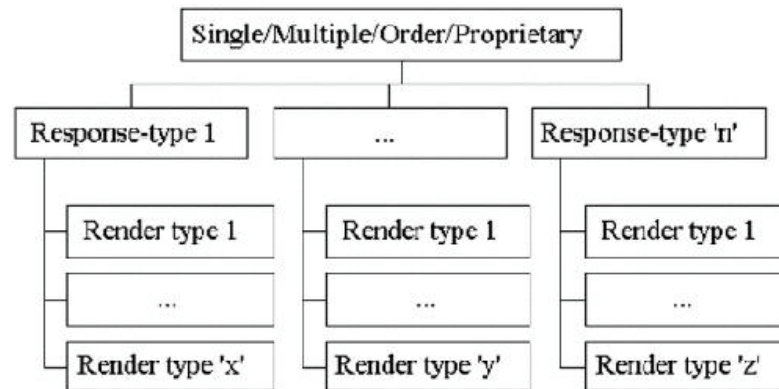


Las taxonomías debajo de las categorías Básica y Compuesta son idénticas. La siguiente subdivisión está basada sobre:

- Dependiente del tiempo: el tiempo tomado en responder es importante y se debe registrar.
- Independiente del tiempo: el tiempo tomado en responder no es significativo;
- El nivel final de la categorización en la figura está basado sobre el número de las acciones requeridas del usuario. Esta categorización da origen a la siguiente tabla:

Response-type	Basic	Composite
Single	A single user response with each item consisting of a single response-type.	A single user response with each item consisting of more than one response-type. Some of the responses will therefore be 'null'.
Multiple	One or more user responses with each item consisting of a single response-type.	One or more user responses with each item consisting of more than one response-type. Some of the responses may be 'null'.
Ordered	One or more user responses with each item consisting of a single response-type, and the order of selection being significant.	One or more user responses with each item consisting of more than one response-type, and the order of the selection being significant. Some of the responses may be 'null'.

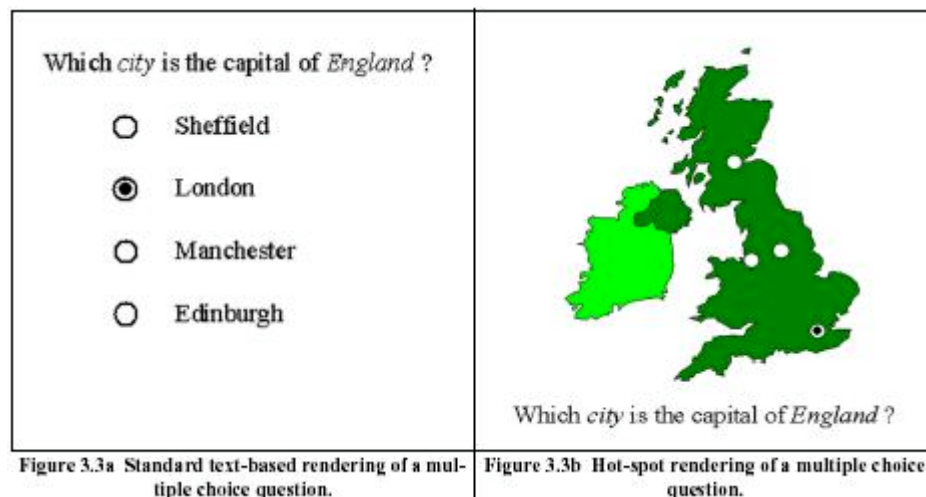
El siguiente nivel de la taxonomía se muestra en la siguiente figura.



Se muestra la relación entre el tipo de respuesta y el diferente formato de presentación, o versión. Para cada una de las clasificaciones hay un número de tipos de respuesta y para cada tipo de respuesta hay uno o más caminos para presentar la selección de la respuesta.

Un ejemplo de esta versión de presentación múltiple se muestra en las figuras 3.3a y 3.3b. En estos dos ejemplos se hace la misma pregunta (“¿qué ciudad es la capital de Inglaterra?”) pero se presenta en dos formatos diferentes:

- Figura 3.3a: usar una lista estándar de respuestas de texto con 'radio buttons';
- Figura 3.3b: usando un gráfico con 'hot-spots';



En las dos representaciones, el usuario debe realizar la misma acción: la identificación de la pieza de la información válida, entre varias opciones posibles. El esquema de clasificación es independiente de los posibles formatos de representación. Esto significa que:

- Se pueden añadir nuevos tipos de respuesta sin alterar la clasificación forme proyectos;
- La representación es independiente de los planes de clasificación;



- Las acciones del usuario se representan exactamente reflejando los flujos de datos requeridos;

TIPOS DE RESPUESTA

Los objetivos primarios de definir los tipos de respuesta, son establecer unas convenciones de vocabulario y convención de nombres y establecer las necesidades de estructura de datos fundamentales.

Tipos de Respuesta Básicos

Los tipos de respuesta básicos explícitamente soportados por esta especificación se muestran en la siguiente tabla. Esta tabla muestra la relación entre los tipos de respuesta, los tipos de pregunta más familiares y los esquemas de clasificación Single/Múltiple/Ordered de esta especificación.

Response-type	Data Structure	Rendering Formats		
		Single	Multiple	Ordered
Logical Identifier (LID)	The response-type identity or list of identities. The order of the list is first choice, second choice, etc.	Multiple choice True/false Slider	Multiple response	Order objects Connect-the-points Match object Order object Drag object Drag target
X-Y Co-ordinates (XY)	The 'x-y' co-ordinates of the centre of the object for each response identity or a list of 'x-y' co-ords. The order of the list is first choice, second choice, etc.	Image hot spot	Order objects	Connect-the-points
String (STR)	The typed string for each response identity.	Fill-in-blank Select text Short answer Essay		
Numerical (NUM)	The entered number for each response identity.	Fill-in-blank Slider		
Logical Groups (GRP)	The response identity and group identity tuples for each matched set of objects.	Match objects Drag object Drag target	Match objects Drag objects Drag targets	Match objects Order objects

Los cinco tipos de respuesta se convierten en los objetos del modelo de información. Las tres categorías single/multiple/ordered se convierten en atributos de la clase básica tipo de respuesta. Un rango de los tipos de representación se puede aplicar entonces a estos tipos de respuesta.

Los tipos de pregunta soportados por esta especificación son:

- Verdadero/Falso:** pregunta con varias posibilidades con respuestas "verdadero o falso", "de acuerdo o desacuerdo", etc.
- Con varias posibilidades:** pregunta con varias posibilidades con unas de las elecciones disponibles identificada por la identidad de respuesta;
- Respuesta múltiple:** pregunta con varias posibilidades con una o más de las elecciones identificadas por las identidades de respuesta;



- **Imagen hot-spot (IHS):** el tipo de respuesta es la ubicación en un gráfico identificada por las coordenadas de "y de x" del punto de la selección;
- **Rellena los huecos (FIB):** entrada formateada para texto o números integer/decimal/scientific identificada por la información introducida. El tipo de respuesta es o un string o un número integer/decimal/scientific respectivamente para cada respuesta;
- **Texto seleccionado:** la identificación del texto en los párrafos o listas presentados. La respuesta es el string identificador o el mapeo a un identificador lógico;
- **Cursor:** la selección de un número entero o cantidad real de un mínimo y un máximo predefinido con un incremento establecido. El tipo de respuesta es un número real o entero o un mapeo a un identificador lógico;
- **Objeto que se arrastra:** estos objetos se mueven en las localizaciones predefinidas del objeto. El tipo de respuesta es identificado por el emparejamiento del objeto fuente con las identidades de objeto objetivo;
- **Objetivo que se arrastra:** los objetos se mueven al objeto objetivo. El tipo de respuesta es identificado por el emparejamiento del objeto objetivo con las identidades de los objetos que han sido movidos;
- **Objetos ordenados:** la reordenación del texto o líneas de texto o un conjunto de objetos revueltos, se tienen que mover a localizaciones predefinidas en una página. La respuesta es identificada por el conjunto de las coordenadas "x-y" de las identidades de respuesta o por su grupo lógico;
- **Objetos relacionados:** los objetos de cada lista se tienen que agrupar juntos. El tipo de respuesta es una n-tupla del conjunto de ítems relacionados, para cada objeto se registra su identidad de respuesta y número de grupo;
- **Unir los puntos:** la conexión ordenada de un conjunto de puntos. El tipo de respuesta es un conjunto de coordenadas "x-y" o LIDs;

Todos estos tipos de pregunta pueden soportar los cinco tipos de respuesta listados en la Tabla del apartado anterior.

Tipos de Respuesta Compuestos

Son las respuestas en que el usuario tenga que dar contestación a dos o más tipos de respuesta básicos (estos tipos básicos pueden ser los mismos). Los ejemplos de dos tipos de respuesta compuestos se muestran en las figuras 3.4 (basaba en varios tipos de respuesta de mismo tipo) y 3.5 (basado en dos tipos de respuesta diferentes, con varias posibilidades y FIB).

Identify the odd one out from each list ?

Sunday	Monday	Tuesday	Wednesday
<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Litre	Pint	Mile	Gallon
<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Inch	Yard	Furlong	Pound
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Year	Decade	Score	Century
<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Figure 3.4 Composite response-type based upon the same basic response-types.

Which *city* is the capital of *England* and name another city in England ?

- ☐ Sheffield
- ☒ London
- ☐ Manchester
- ☐ Edinburgh

Another city:

Figure 3.5 Composite response-type based upon the different basic response-types.

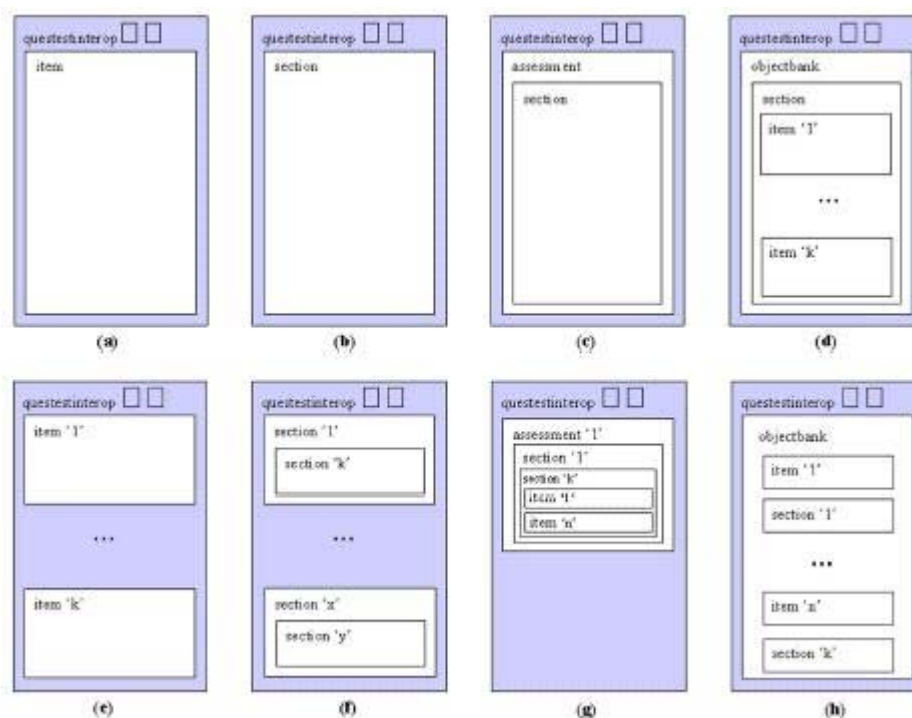
14.8.4. Modelo de Información Básico

EVALUACIONES, SECCIONES, ITEMS Y BANCOS DE OBJETOS

La especificación IMS QTI se basa en cuatro objetos de datos principales:

- **Ítem:** el objeto intercambiable más pequeño dentro del QTI-XML. Un ítem es mas que una pregunta, ya que contiene la pregunta, las instrucciones de representación, el proceso de respuesta que debe ser aplicado a las respuestas del participante, la realimentación que debe ser presentada y los meta-datos que describen el ítem.
- **Evaluación:** una evaluación es equivalente a un examen. Contiene la colección de ítems que se usan para determinar el nivel que un participante tiene en una asignatura en particular. La evaluación contiene todas las instrucciones necesarias para almacenar la puntuación individual en cada ítem y así producir la puntuación final.
- **Sección:** se usa para construir objetos de evaluación jerárquicos. Puede contener una o más secciones. Se usa para cubrir 2 tipos de necesidades:
 1. Para representar diferentes agrupaciones como está definido por el paradigma de educación adecuado.
 2. Para limitar la extensión de las instrucciones de secuencia y para controlar las distintas formas en las que se pueden construir;
- **Banco de objetos:** colección de artículos, secciones o una mezcla de artículos y secciones. Un banco de objetos tiene su propio identificador único y meta-datos. Un banco de objeto se usa para contener la base de datos de objetos de evaluación que puede usarse para construir evaluaciones.

ESTRUCTURAS DE DATOS INTEROPERABLE



Esta representación muestra la relación entre los elementos de ASI y se resume de la siguiente forma:

- Una evaluación consiste de al menos una sección (c);
- Una sección puede contener otras secciones (b) y (f);
- Una sección puede contener unos o más ítems (d) y (h) - esto permite la definición de sección nula;
- Un banco de objetos puede consistir en solo ítems, solo secciones, o una mezcla de ítems y secciones.

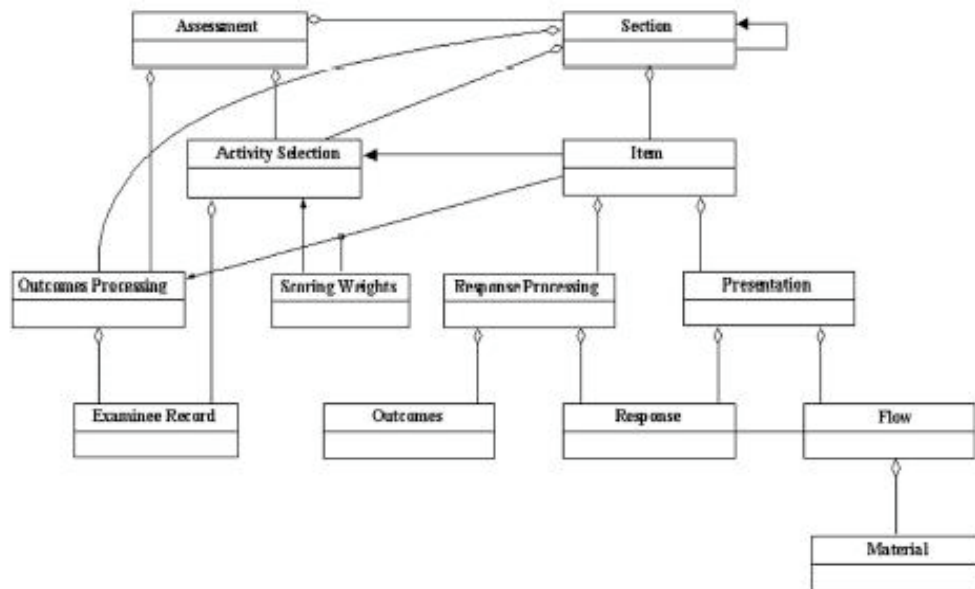
La estructura de datos puede usarse para importar/exportar estructuras de datos que consisten en:

- La estructura de datos puede usarse para importar/exportar estructuras de datos que consisten en:
 - Una o más secciones solo (b) y (f);
 - Uno o más ítems solo (a) y (la e);
 - Una evaluación puede o no contener más de una sección (c) y (g);
 - Una sección puede o no contener ítems (b), (c), (f) y (g);
 - Un banco de objetos (d) y (h);

Los bancos de objetos son intercambiados definiendo un tipo de paquete QTI, por ejemplo, un conjunto de datos similares almacenados en el elemento <questesrinterop>.

MODELO DE OBJETO

El modelo de objeto conceptual para la QTI se muestra en la siguiente figura.



Los objetos en este modelo y sus comportamientos claves son:

- **Evaluación:** el objeto que representa la estructura de datos *evaluación* ;
- **Sección:** el objeto que representa la estructura de datos *sección* ;
- **Item:** el objeto que represente la estructura de datos *ítem* ;
- **Selección de actividad:** la selección de la siguiente actividad determinada por el progreso y resultados obtenidos hasta el momento;
- **Proceso de resultados:** la reconciliación de todas las salidas de evaluación para producir una evaluación completa;
- **Pesos de puntuaciones:** los pesos de las puntuaciones que se asignan a los resultados de salida del proceso de respuesta;
- **Proceso de respuesta:** el proceso y evaluación de las respuestas de usuario;
- **Presentación:** la versión del contenido y posibles respuestas;
- **Registro de examinando:** el conjunto de los resultados que son la salida del proceso completo. Esto podría ser un registro 'life-long' en el que almacenar el progreso histórico del individuo;

- **Resultados:** el conjunto de resultados que van a ser evaluados por el objeto de proceso de respuesta. Éstos determinan las métricas que se deben aplicar las evaluaciones de respuesta;
- **Respuesta:** las respuestas que son suministradas por el usuario de los ítems;
- **Flujo:** estructura que define las relaciones de bloque entre los diferentes componentes material;
- **Material:** el contenido que debe mostrarse.

TIPOS DE RESPUESTA

Dentro de un ítem hay un conjunto complejo de estructuras de datos basadas en los tipos de respuesta. La información devuelta por los tipos de respuesta básicos puede definirse en términos de listas de objetos, denominados:

- Logical identifier (LID)
 - Single = {identifier}, {duration}
 - Multiple = {identifier, identifier, ..., identifier, duration}, {duration}
 - Ordered = {event_1_identifier, event_2_identifier, ..., event_k_identifier}, {duration}
- X-Y co-ordinates (XY)
 - Single = {{identifier, xcoord, ycoord}}, {duration}
 - Multiple = {{identifier, xcoord, ycoord}, {identifier, xcoord, ycoord}, ..., {identifier, xcoord, ycoord}}, {duration}
 - Ordered = {{event_1_identifier, xcoord, ycoord}, {event_2_identifier, xcoord, ycoord}, ..., {event_k_identifier, xcoord, ycoord}}, {duration}
- String (STR)
 - Single = {{identifier, string}}, {duration}
 - Multiple = {{identifier, string}, {identifier, string}, ..., {identifier, string}}, {duration}
 - Ordered = {{event_1_identifier, string}, {event_2_identifier, string}, ..., {event_k_identifier, string}}, {duration}
- Numerical
 - Single = {{identifier, number}}, {duration}
 - Multiple = {{identifier, number}, {identifier, number}, ..., {identifier, number}}, {duration}
 - Ordered = {{event_1_identifier, number}, {event_2_identifier, number}, ..., {event_k_identifier, number}}, {duration}
- Logical group (GRP)
 - Single = {{{identifier, groupid}, {identifier, groupid}}}, {duration}
 - Multiple = {{{identifier, groupid}, {identifier, groupid}, ..., {identifier, groupid}}, {{identifier, groupid}, {identifier, groupid}, ..., {identifier, groupid}}, ..., {{identifier, groupid}, {identifier, groupid}}, ..., {identifier, groupid}}, {duration}
 - Ordered = {{{event_1_identifier, groupid}, {event_1_identifier, groupid}, ..., {event_1_identifier, groupid}}, {{event_2_identifier, groupid}, {event_2_identifier, groupid}, ..., {event_2_identifier, groupid}}, ..., {{event_k_identifier, groupid}, {event_k_identifier, groupid}, ..., {event_k_identifier, groupid}}}, {duration}

CONTENIDO

Flujos

Un flujo se define como un conjunto de contenido que va a manejarse por el motor de versión como un bloque lógico, o párrafo. Un flujo puede contener otros flujos y así un sistema complejo de flujos jerárquicos se puede construir. En el caso de la siguiente figura, tenemos dos flujos o bloques.

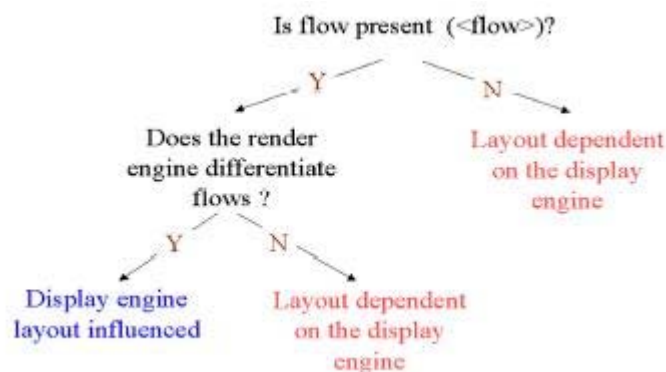
Fill-in-the blanks in this text from
Richard III:

Now is the _____ of our
discontent made glorious _____
by these sons of _____.

Los flujos se pueden usar o no. Cuando los flujos son usados esto será indicado por la presencia del elemento `<flow>` a nivel superior dentro del elemento `<presentation>`. Si cualquier construcción flujo va a ser usada, entonces el elemento `<flow>` debe estar presente dentro del elemento `<presentation>`. Se definen 3 elementos separados para soportar flujos:

- **`<flow>`** se usa para indicar el flujo de nivel superior dentro del elemento `<presentation>`;
- **`<flow_etiqueta>`** se usa para encapsular el elemento `<response_label>`;
- **`<flow_estera>`** se usa para encapsular el elemento `<material>`;

En todos los casos el elemento puede ser recursivo, por ejemplo, `<flow_mat>` dentro de `<flow_mat>` y las reglas de bloque deber ser definidas e implementadas consistentemente.





14.8.5. Descripción Conceptual de los Objetos de Datos

Las tablas proporcionan una descripción conceptual de los elementos en los objetos de datos. Las columnas a las que se refieren son:

No: el número del elemento. Un elemento puede estar compuesto por sub-elementos. La numeración del esquema refleja estas relaciones

Name: el nombre descriptivo del elemento

Explanation: descripción del elemento.

Required: indica si el elemento es requerido:

- M= elemento obligatorio que debe ser incluido en el objeto de datos, si el elemento al nivel mas alto esta incluido
- C= elemento condicional. Existe o no dependiendo de los valores de otros elementos
- O= elemento opcional

Multi: multiplicidad del elemento

- Blank: instancia simple
- Number: máximo numero de veces que el elemento se repite
- n: sin limite, permitidas múltiples ocurrencias

Type: descripción de las reglas de formación del elemento, del conjunto de entradas válidas para los datos y esquemas. El conjunto de valores como están definidos en el Dominio. La lista de valores no puede extenderse para incluir valores que no están definidos en la especificación. Si hay necesidad de valores no incluidos en el dominio, entonces, la extensión se puede hacer definiendo un nuevo elemento bajo el elemento Extension que es parte de cada definición de objeto de dato. Type incluye la máxima longitud del contenido de los elementos:

- ID= elemento usado para identificar de manera única un objeto
- Code= valor del elemento que se obtiene de una lista de códigos
- Descripción= elemento descriptivo, lenguaje humano
- Flag= flag binario
- Enumerated= lista de opciones no numéricas predefinidas, por ejemplo: la lista definitiva de objetos.

En todos los campos se puede usar el conjunto de caracteres especificado por ISO 10646.

Nota: información descriptiva adicional sobre el elemento.



La descripción de los objetos de datos tiene 7 sub-secciones:

- Questestinterop - es la estructura raíz del QTI y contiene el conjunto de evaluaciones, secciones, ítems y el banco de objetos
- Assessment - describe la estructura de datos evaluación
- Section - describe la estructura de datos sección
- Item - describe la estructura de datos ítem
- Common elements - describe los elementos usados en más de una de las estructuras de datos anteriores
- Common attributes - describe los atributos usados en más de una de las estructuras de datos anteriores
- Meta-data - describe los objetos metadata específicos de QTI usados en los objetos de datos Assessment, Section e Item

Tablas...

- [<questestinterop> Data Object](#) el objeto Questestinterop es la raíz de la estructura de la especificación QTI. Puede contener uno o más evaluaciones, secciones o ítems
- [<assessment> Data Object](#) un objeto Assessment contiene toda la información para hacer significativo el uso de Ítems individuales, es decir, a parte de las secciones, el objeto incluye las relaciones entre secciones, el procesamiento del grupo de evaluación y la correspondiente realimentación.
- [<section> Data Object](#) un objeto Section contiene toda la información para hacer significativa la agrupación de ítems, es decir, aparte de los ítems, el objeto incluye la relación entre los ítems y el criterio de selección de éstos.
- [ITEM Data Object](#) un objeto Item contiene toda la información de la presentación de una pregunta y el procesamiento para la respuesta del usuario. La estructura item incluye la pregunta actual y su formato de presentación, el rango de posibles respuestas, las formas en las que las respuestas deben ser procesadas, las posibles soluciones y sugerencias para el item y la realimentación.
- [<questestinterop> Data Object](#) el objeto Questestinterop es la raíz de la estructura
- [Common Data Objects \(Elements\)](#) describe los objetos de datos comúnmente usados con los objetos Objectbank, Assessment, Section e Item



- [Meta-Data Object](#) describe los objetos meta-data inicialmente disponibles a los objetos Item. Ninguna de estas estructuras puede usarse ya que están se han deprecado en favor del vocabulario específico de meta-datos QTI (ver Sección 6)
- [Common Data Objects \(Attributes\)](#) describe los atributos comúnmente usados con los objetos Objectbank, Assessment, Section e Item

14.8.6. Ver evaluación en Internet

14.9. IMS LEARNING DESIGN SPECIFICATION

14.9.1. Introducción

La tarea clave del Modelo de Información del Diseño de Aprendizaje IMS, es el desarrollo de un entorno de trabajo que soporte tanto diversidad pedagógica como innovación mientras promueve el intercambio e interoperabilidad de los materiales e-learning.

Es un estándar muy reciente, cuya primera versión de prueba (Versión 1.0 Public Draft Specification) surgió el 18 de Octubre del 2002, y no ha sido hasta Febrero-Marzo del 2003 cuando se ha conocido su versión final para dar un entorno de trabajo que tenga los elementos necesarios para describir cualquier diseño de un proceso enseñanza-aprendizaje. El método empleado para conseguir esto se llama **Learning Design (Diseño de Aprendizaje)** y consiste en dar un conjunto de **Actividades** (de resolución de problemas, de búsqueda, de discusión, de valoración, etc.), a los diferentes Roles (o papeles) del alumno, para que las pueda ir completando en un cierto orden y así ir alcanzando los objetivos de aprendizaje propuestos en dicho proceso enseñanza-aprendizaje.

Cada "Learning Design" es un Método prescribiendo varias Actividades para los diferentes Roles del alumno y de la plantilla en un cierto orden.

Cada actividad se refiere a una colección de servicios y objetos específicos (llamados ENTORNO) necesarios para hacer la actividad.

Hay tres niveles de implementación:

El Diseño de aprendizaje (Learning Design) especifica tres niveles de implementación o condescendencia que estarán trazados, cada uno, en un esquema XML separado. Para reflejar esto, este documento se divide en tres:

- **Learning Design Level A:** Contiene todo lo descrito hasta ahora. Así contiene todo el núcleo del vocabulario que se necesita para dar soporte a la gran diversidad pedagógica.
- **Learning Design Level B:** Añade Propiedades y Condiciones al nivel A, las cuales permitirán una personalización y unas secuencias e



interacciones más elaboradas basadas en el portafolios o carpeta del alumno.

- **Learning Design Level C:** Añade Notificación al nivel B.

La aproximación o propuesta hecha en esta especificación no es, por tanto, definir un esquema único y largo con un núcleo de elementos mandatorios y otros elementos opcionales, sino definir un núcleo completo lo más sencillo como sea posible y entonces definir dos niveles de extensión que capturen rasgos y comportamientos más sofisticados.

Los documentos de instancias implementados según esta especificación del "Learner Design", no necesitan implementar cada elemento. Los sistemas deben implementar toda la especificación al nivel expuesto y deben ejecutar todas las instancias de ese nivel sean cual sean las opciones elegidas. Las instancias que satisfacen esta especificación deben ser validadas por un compilador usando los esquemas XML dados.

Otras especificaciones:

El Diseño de Aprendizaje (Learning Design) puede ser considerado como una capa integrable con muchas especificaciones existentes. La especificación IMS Learning Design hace uso, incluye o es extendida con las siguientes especificaciones, a través del mecanismo XML de los Namespaces:

- IMS Content Packaging.
- IMS Simple Sequencing.
- IMS/LOM Meta-Data.
- IMS Question and Test Interoperability.
- IMS Reusable Competency Definition.
- IMS Learner Information Package.
- IMS Enterprise.
- Es posible incluir contenido SCORM.

Alcance y contexto:

Este documento es la "IMS Learning Design Specification" (Especificación del Diseño de aprendizaje) y será usado como base para la producción de los siguientes documentos:

- IMS Learning Design XML Bindings (obligaciones XML) para los niveles A, B y C.
- IMS Learning Design Best Practice and Implementation Guide.



Estos documentos juntos constituyen la "IMS Learning Design Specification" (Especificación del Diseño de aprendizaje).

El objetivo de la Especificación del Diseño de Aprendizaje (Learning Design Specification) es dar un marco de trabajo que contenga elementos, que puedan describir cualquier diseño de un proceso enseñanza-aprendizaje de un modo formal.

Más específicamente cumple los siguientes requisitos:

- **Compleitud:** La especificación debe describir el proceso en una unidad de aprendizaje, incluyendo referencias a los objetos digitales y no digitales y a los servicios necesarios.
- **Flexibilidad Pedagógica:** debe expresar la funcionalidad y significado pedagógico de los diferentes datos en el contexto de una unidad de aprendizaje. No debe prescribir u obligar a ninguna aproximación pedagógica específica.
- **Personalización:** Debe describir aspectos de personalización que el contenido y actividades de una unidad de aprendizaje puedan ser adaptados y basados en las preferencias, portafolios, conocimientos previos, necesidades educativas y situaciones circunstanciales de los usuarios.
- **Formalización:** La especificación debe ser un modo formal para que el procesamiento automático sea posible.
- **Reproducibilidad:** La especificación debe ser abstraída de forma que la ejecución repetida de diferentes escenarios con distintas personas sea posible.
- **Interoperabilidad:** Debe soportar interoperabilidad de diseños de aprendizaje.
- **Compatibilidad:** Usa las especificaciones y estándares disponibles cuando sea posible, principalmente el IMS Content Packaging, IMS Question and Test Interoperability, IMS/LOM Meta-Data e IMS Simple Sequencing.
- **Re-usabilidad:** La especificación debe hacer posible identificar, aislar, descontextualizar e intercambiar útiles artefactos de aprendizaje y rehusarles en otros contextos.

Este modelo de información describe un modelo para el Diseño de aprendizaje (Learning Design) que contiene tres componentes principales, que se explicarán con detalle más adelante.



- Un Modelo conceptual que representa el vocabulario y las relaciones entre los conceptos, además de la relación con el IMS Content Packaging.
- Un Modelo de Información que describe los elementos del IMS Learning Design para los niveles A, B y C respectivamente.
- Un Modelo de comportamiento que describe un conjunto de comportamientos en tiempo de ejecución que los sistemas entregados deben implementar.

14.9.2. Modelo Conceptual

El modelo conceptual está expresado como un conjunto de modelos de clases UML y una definición del vocabulario usado. Todos los modelos se dan antes de que el vocabulario sea definido. Hay tres modelos básicos:

- Modelo de Agregación.
- Modelo de estructura.
- Modelo que representa la integración del diseño de aprendizaje (Learning Design) con el IMS Content Package, para obtener lo que se ha llamado Unidad de Aprendizaje "Unit of Learning".

Niveles de agregación semántica en el diseño de aprendizaje:

La notación empleada en el siguiente diagrama es conforme a la versión 1.4 del UML, y representa sólo relaciones de agregación (incluyendo composiciones) y las especializaciones de las clases abstractas (tipos).

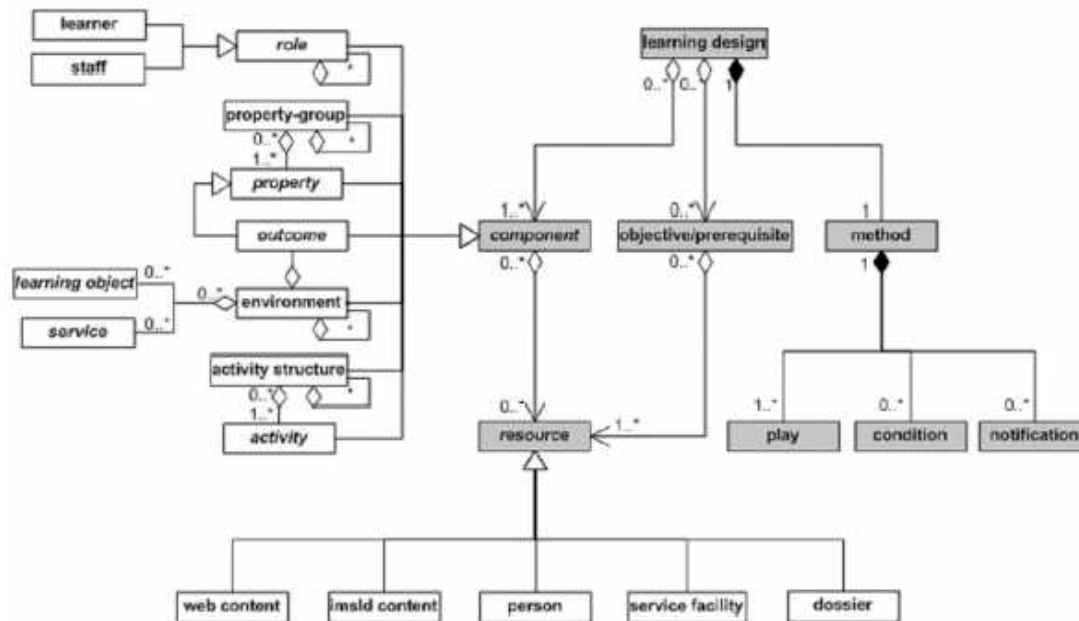


Figure 2.1 - Semantic aggregation levels in the Learning Design level C specification (grey coloring is only used to increase the readability).

El modelo muestra que el diseño de aprendizaje (Learning Design) da una vista semántica de una colección de recursos, por un lado, y por el otro integra un método, especificando los aspectos dinámicos del diseño de aprendizaje.

Este modelo muestra tres niveles de agregación semántica (Las capas horizontales de color gris). El nivel semánticamente más alto es el learning design, agrega una colección de componentes, objetivos/ prerequisites y un método. El nivel más bajo lo representan resource (recurso), escena, condición y notificación. Los recursos son agregados dentro de componentes y objetivos/prerequisites. Las escenas, condiciones y notificaciones son agregadas al método.

Un componente (component) puede ser uno de los siete tipos siguientes (que aparecen en el diagrama): papel (role), propiedad de grupo (property-group), propiedad (property), resultado (outcome), ambiente (environment), estructura de actividad (activity structure), actividad (activity). Excepto el resultado (outcome), estos tipos son elementos en el Modelo de Información del Diseño de Aprendizaje (LD Information Model).

Un recurso (resource) puede ser uno de los siguientes tipos: contenido Web (Web content), contenido imslid (imsld content), persona (person), facilidades de servicio (service facility), dossier.

Algunos tipos específicos de componentes son obligatorios para determinados tipos de recursos. El momento en el que el recurso es necesitado u obligatorio en el diseño de aprendizaje varía:



Component	Binds to Resource Type	Moment of binding
Role	Person	during instantiation and during runtime
Objective/prerequisite	Web content	during design, instantiation or during runtime
Objective/prerequisite	Imstd content	during design
Property	Dossier	during instantiation
Learning object	Web content	during design, instantiation or during runtime
Learning object	Imstd content	during design
Service	Service facility	during design, instantiation or during runtime
Activity	Web content	during design, instantiation or during runtime
Activity	Imstd content	during design

Ejemplo: la descripción textual de un objetivo de aprendizaje (learning objective) puede ser escrita durante el diseño y entregada como un recurso fijo con el diseño de aprendizaje. Sin embargo también puede darse una URL (Universal Resource Locator) absoluta a un lugar donde el fichero puede ser editado en cualquier momento.

El modelo también muestra que los componentes, objetivos/prerrequisitos y recursos son independientes del diseño de aprendizaje (learning design), por lo que pueden ser referenciados y usados en muchos otros diseños de aprendizaje.

Estructura conceptual del diseño de aprendizaje:

En este modelo, el énfasis está en las relaciones funcionales entre las clases. En el siguiente diagrama se muestra otra vista conceptual del diseño de aprendizaje:

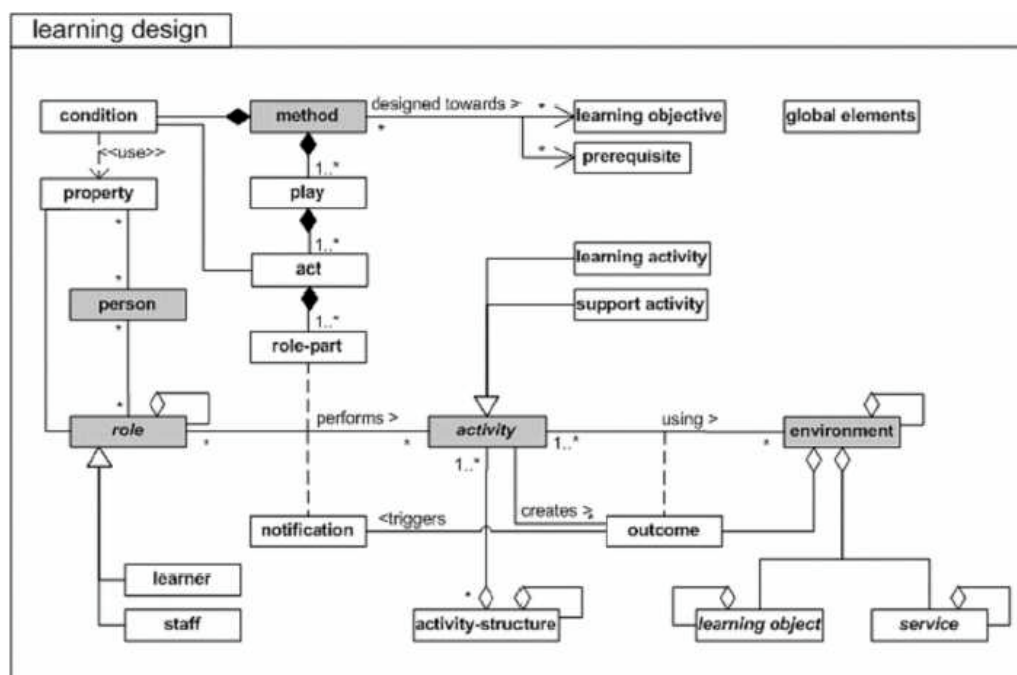


Figure 2.2 -Conceptual model of overall Learning Design structure Level C (gray coloring is only used to increase the readability).



El concepto central de la especificación del Diseño de Aprendizaje (Learning Design specification), es que, a pesar de el acercamiento pedagógico, una persona (person) obtiene un papel (role) en el proceso de enseñanza-aprendizaje, que típicamente será un rol de alumno (learner) o de profesorado (staff). En este papel o rol trabaja hacia ciertos resultados ejecutando unas actividades de soporte y/o aprendizaje (learning/support activities) más o menos estructuradas en un ambiente (environment). El entorno consiste en que los objetos de aprendizaje (learning objects) y los servicios (services) apropiados sean usados durante la ejecución de las actividades. Qué rol coger, con qué actividades y en qué momento está determinado por el modelo de información, pero algunos sólo existen a nivel conceptual (persona, resultado).

Un método puede, en el nivel B, contener condiciones, es decir, reglas if-then-else que refinen más la visibilidad de las actividades y entidades de entorno para las personas y roles, definiendo expresiones Booleanas en sus propiedades. Una propiedad puede ser agrupada en un grupo de propiedades.

Una notificación es lanzada por un resultado y puede hacer disponible una nueva actividad para que rol se ejecute. La persona que recibe la notificación no es necesariamente la misma que crea la salida. Por ejemplo, cuando un estudiante completa una actividad (un resultado), entonces otro estudiante o el profesor pueden ser notificados y poner otra actividad como consecuencia.

Las estructuras de actividad agregan un conjunto de actividades relacionadas a una estructura sencilla, que puede ser asociada a un rol en un role-part. Una estructura puede modelar una secuencia (el rol debe completar las diferentes actividades de la estructura en el orden dado) o una selección de actividades (el rol puede seleccionar un número de actividades de entre las dadas en la estructura de actividad). Las estructuras de actividad (Activity-Structures) pueden referenciar también a otras estructuras de actividad y a unidades de aprendizaje externas, permitiendo definir estructuras elaboradas si son requeridas.

Adicionalmente, la especificación IMS Simple Sequencing puede ser incluida en las estructuras de actividad para permitir comportamientos secuenciales más elaborados. Cuando se hace esto, las estructuras de actividad son obligatoriamente jerarquías de árbol.

Para permitir a los usuarios el agrupar y ver las propiedades del nivel B del contenido que se les presenta, los llamados elementos globales (global elements) son presentados en el modelo. Estos elementos son designados para ser incluidos en cualquier esquema de contenido a través de los Namespaces. Y para permitir interacciones en tiempo de ejecución con los usuarios finales, los elementos globales del diseño de aprendizaje se dan separadamente al nivel B, el cual puede ser introducido o llamado mediante los namespaces en un esquema de contenido basado en XML.

UNIDAD DE APRENDIZAJE = IMS CONTENT PACKAGE + IMS LEARNING DESIGN:

El principal uso del Diseño de Aprendizaje IMS (IMS Learning Design) es modelar unidades de aprendizaje (units of learning), para ello se incluye un IMS Learning Design en un paquete de contenido, preferiblemente -pero no necesariamente- un Paquete de Contenido IMS (IMS Content Package). En esta especificación se asume que el Diseño de Aprendizaje IMS se usa con un Paquete de Contenido IMS para modelar unidades de aprendizaje.

Los Paquetes de Contenido IMS describen sus contenidos en un documento XML llamado el paquete de manifiesto (package manifest). Este Manifiesto puede incluir "vistas" estructuradas en los recursos contenidos en ese paquete; cada vista está descrita como una jerarquía de ítems llamados una "organización" (organization). Cada ítem se refiere a un Recurso (Resource) que, uno tras otro, pueden referirse a un fichero físico dentro del paquete. También puede referirse a un recurso externo. La siguiente figura representa el modelo conceptual IMS Content Package entero:

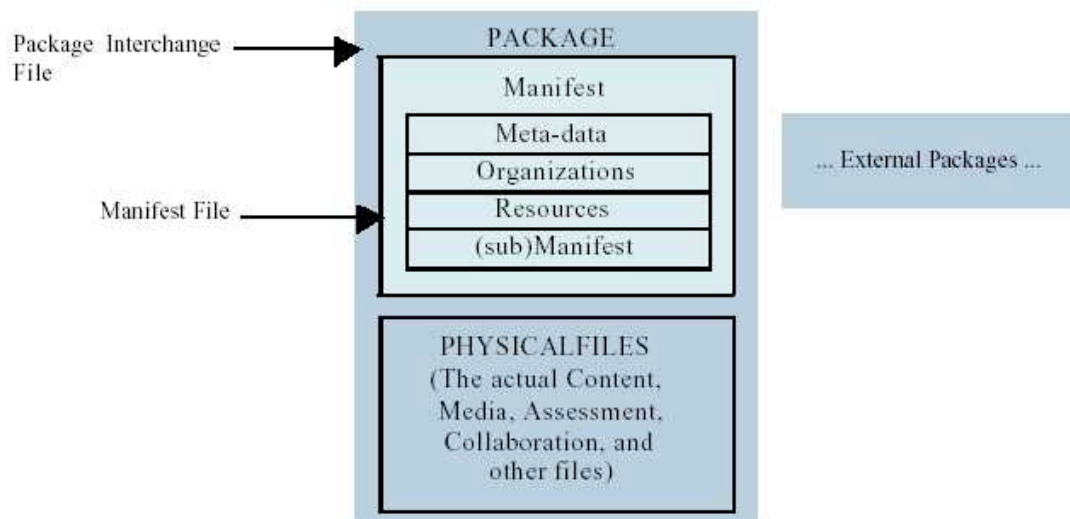
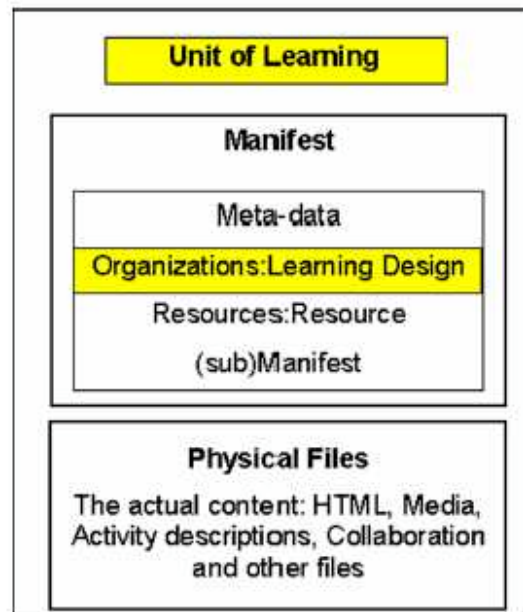


Figure 2.1 IMS Content Packaging scope.

El Manifiesto (Manifest) es la estructura de información definida en la especificación del Content Packaging. Esta información está contenida dentro de un paquete como un fichero XML con un nombre fijo, predefinido "imsmanifest.xml"

La integración de un Diseño de Aprendizaje (Learning Design) en la Estructura del Paquete de Contenido (Content Package Structure) está reflejada en la siguiente figura:



The structure of a Unit of Learning, composed by including an IMS Learning Design within the Organizations part of IMS Content Packaging.

Para crear una unidad de aprendizaje, se integra el Diseño de Aprendizaje IMS (IMS Learning Design) en un Paquete de Contenido IMS (IMS Content Package) incluyendo el elemento diseño de aprendizaje (learning design) como otro tipo de organización en el elemento organizaciones (organizations), usando el namespace estándar para el Diseño de Aprendizaje (Learning Design). Los elementos diseño de aprendizaje (learning design) son incluidos como sigue (ignorando atributos y elementos irrelevantes):

```
<manifest>
  <metadata/>
  <organizations>
    <learning-design xmlns="[standard-namespace-for-learning-design]">
      [add learning design elements here]
    </learning-design>
  </organizations>
  <resources/>
</manifest>
```

Las cursivas deben ser rellenadas con el namespace y los elementos apropiados respectivamente.

En un paquete que incluye un elemento learning design, el elemento de organización opcional dentro de organizaciones es ignorado y sólo el elemento learning design es leído por el sistema en tiempo de ejecución. Cuando se deseen otros elementos organización, pueden ser incluidos en sub-manifiestos, como sub-paquetes pueden ser agregados de la misma forma que en paquetes de contenido normales.



14.9.3. Vocabulario Conceptual del Diseño de Aprendizaje (Learning Design)

Daremos los términos conceptuales básicos de la especificación del Diseño de Aprendizaje:

- **Unidad de aprendizaje (Unit of Learning):** Es un término abstracto usado para referirse a alguna pieza delimitada de educación o entretenimiento, tales como un curso, un módulo, una lección, etc. Representa algo más que una colección de recursos ordenados para aprender, incluye una variedad de actividades (actividades de resolución de problemas, de búsqueda, de discusión, de valoración, etc.) valoraciones, servicios y facilidades de soporte dadas por parte de los profesores, entrenadores y otros miembros del profesorado. Qué actividades, recursos, roles y flujos de trabajo se den, dependen del diseño de aprendizaje (learning design) en la unidad de aprendizaje.

Una unidad de aprendizaje puede ser modelada como un Paquete de Contenido IMS (IMS Content Package) incluyendo un Diseño de Aprendizaje IMS (IMS Learning Design) en el paquete.

Un Paquete de Contenido IMS se llama Unidad de Aprendizaje si y sólo si incluye un elemento diseño de aprendizaje IMS válido en la parte organizaciones del manifiesto del paquete.

Una Unidad de Aprendizaje incluye un manifiesto, un diseño de aprendizaje, recursos, posibles (sub)manifiestos y ficheros físicos.

- **Diseño de aprendizaje (Learning Design):** Es una descripción de un método que permite a los alumnos alcanzar ciertos objetivos de aprendizaje haciendo ciertas actividades de aprendizaje en un cierto orden en el contexto de un cierto entorno de aprendizaje. Un diseño de aprendizaje se basa en los principios pedagógicos del diseñador y en un dominio y variables contextuales específicas (por ejemplo, los diseños para enseñar matemáticas pueden ser distintos que para enseñar lenguaje; para educación a distancia pueden ser distintos que para educación cara a cara).

Los profesores y entrenadores aplican sus propios principios de enseñanza. Para permitir que todos estos diseños diferentes sean incluidos efectivamente en módulos de e-learning, se coge la aproximación de un meta-lenguaje.

El elemento Diseño de Aprendizaje es el elemento raíz de la Especificación del Diseño de Aprendizaje (Learning Design Specification).

- **Objetivos de aprendizaje:(Learning Objectives):** Los tienen que alcanzar los alumnos que completan una unidad de aprendizaje. En el Diseño de Aprendizaje IMS (IMS Learning Design), los diseñadores pueden especificarlos a dos niveles: al nivel global de la unidad de aprendizaje o para cada actividad simple.



- **Prerrequisitos:** Especifican los requisitos generales de entrada para que los alumnos hagan la unidad de aprendizaje. Pueden proveerse a nivel de la unidad de aprendizaje y/o para actividades de aprendizaje individuales.

Los objetivos de aprendizaje y los prerrequisitos pueden describirse usando el formato Definición de Competencias IMS (IMS Competencies Definition), pero también puede referirse a recursos simples (por ejemplo un texto) con una descripción del objetivo de aprendizaje.

- **Componentes:** Son las declaraciones de los diferentes componentes que proporcionan los bloques de construcción (building blocks) para la sección de métodos del diseño de aprendizaje. En el nivel A son: roles, actividades y entornos. En los niveles B y C son: roles, propiedades, actividades y entornos. Los componentes se declaran separadamente al método para permitir duplicación en el método cuando usa el mismo componente más de una vez, (como en una receta de cocina, donde los componentes son la lista de ingredientes y el método son las instrucciones de preparación).
- **Roles o papeles:** Permiten especificar el tipo de participante. Los dos roles básicos son: Alumno (Learner) y Profesorado (Staff). Pueden tener subtipos y ser más especializados.
- **Propiedades:** Sólo están disponibles en el nivel B y C y del Diseño de Aprendizaje. Forman las bases sobre las que se construyen los dossiers y portafolios de usuarios y roles. Son fundamentales para la interacción con el usuario. Hay cinco tipos de propiedades: locales (declaradas en el diseño de aprendizaje), local-personales, local-roles, global-personales y globales (declaradas externamente). Las propiedades se pueden agrupar.
- **Elementos globales:** En los niveles B y C, los elementos globales se dan como una parte separada de la especificación Diseño de Aprendizaje IMS para que los usuarios puedan poner y ver las propiedades durante el proceso enseñanza-aprendizaje. No son parte del árbol de diseño de aprendizaje sino que están diseñados para ser incluidos en cualquier esquema de contenido XML mediante los namespaces. Sin estos elementos no es posible acceder ni establecer propiedades.

Hay cuatro elementos globales: Set property (para cambiar el valor de una propiedad), View-property (muestra el valor de una propiedad seleccionada), Set-property-group y View-property-group (igual pero para un grupo de propiedades).



- **Actividades:** Son los ejercicios o actividades que un role debe seguir en un entorno determinado, están compuestas de objetos de aprendizaje (learning objects) y servicios. También especifican las condiciones de terminación y qué hacer cuando se termina la actividad.

Hay dos tipos: Actividades de Aprendizaje (dirigidas a alcanzar un objetivo de aprendizaje por un individuo particular), Actividades de soporte (facilitan a un role el poder realizar una o mas actividades de aprendizaje, habrá una por cada usuario del role que se soporte). Además las actividades se pueden agregar a una Estructura de Actividad, la cual da el mecanismo para estructurar las actividades y unidades de aprendizaje referenciadas.

- **Entorno:** Las actividades tienen lugar en lo que se llama el entorno, que es una colección de objetos de aprendizaje, servicios y sub-entornos.
- **Objeto de aprendizaje:** Cualquier recurso digital o no digital reproducible y direccionable usado para ejecutar actividades de aprendizaje o de soporte. En el IMS Content Packaging están representados como recursos: páginas Web, editores, calculadora, libros de texto...
- **Servicio:** Las facilidades de servicio son recursos que no pueden estar en una URL en el tiempo de diseño. Son usadas durante el aprendizaje y la enseñanza, por ejemplo, un foro de discusión o cualquier otra facilidad de comunicación. Hay diferentes tipos: servicio de envío de correo, de conferencia, de monitor y de búsqueda indexada.
- **Método:** Contiene dos partes centrales de la especificación del diseño de aprendizaje: la escena o representación y sus condiciones, junto con algunas declaraciones.
- **Escena o representación (Play):** Se especifica qué roles ejecutarán qué actividades y en qué orden (se puede comparar a las obras de teatro). Cuando se lee un diseño de aprendizaje (learning design), básicamente uno lee la representación, que se modela de acuerdo a una obra teatral con actos y roles. Es una secuencia de actos. En cada acto, diferentes actividades se dan para diferentes roles y estas son ejecutadas en paralelo. Cuando el acto esta completo, empieza el siguiente y así hasta que se cumplen los requisitos de completitud para el diseño de aprendizaje.
- **Condiciones:** Disponibles en los niveles B y C del Diseño de Aprendizaje IMS. Se usan junto a las propiedades para refinar y añadir facilidades de personalización el diseño de Aprendizaje. Tienen la forma: IF-THEN



- **Notificación:** Disponible en el nivel C del seño de Aprendizaje IMS. Mediante la notificación es posible enviar un mensaje a un role o asignar nuevas actividades basadas en eventos.
- **Item:** Es una descripción de una actividad. Cuando un componente, un objeto de aprendizaje o un prerequisite necesitan un recurso, un elemento ITEM se usa de forma similar a como se usa en la parte organización del IMS Content Packaging.

14.9.4. Modelo de Información

El modelo de información para el nivel A, B y C será presentado en el siguiente formato:

- El modelo conceptual UML, derivado de la figura que veíamos en el apartado Estructura Conceptual del Diseño de Aprendizaje llamada "learning design", mostrando sólo los elementos que son apropiados para este nivel. Los ítems no usados son borrados.
- Diagramas de árbol representando partes sucesivas de los árboles aprendizaje-diseño y elementos-globales.
- Tablas de información de partes de la estructura de información. Los diagramas y tablas son desempaquetadas sucesivamente. Cuando el mismo elemento ocurre más de una vez en una tabla, la información no se repite, sino que se referencia con "ver arriba".

Los diagramas usan el siguiente formato:

- Sólo se muestran elementos (no atributos).
- Los diagramas de árboles, para ser leídos de izquierda a derecha. Un elemento de la izquierda contiene a los elementos a mano derecha. El elemento más a la izquierda es la raíz del árbol.
- Sólo se representan dos árboles en todo el documento: un árbol con la raíz "learning design" y un pequeño árbol con raíz "global elements".
- Una relación OR en los diagramas se representa con <.
- Una relación AND en los diagramas se representa con [.
- Un * significa que el elemento aparece cero o más veces en el contenedor.
- Un + significa que el elemento aparece una o más veces en el contenedor.
- Una ? significa que el elemento es opcional.



- Cuando ninguno de los símbolos anteriores se pone antes del nombre del elemento, este aparece exactamente una vez.

Las tablas describen los elementos y atributos del diagrama bajo el cual están colocados. Estas tablas tienen el siguiente formato:

- **No:** El número de elemento en la jerarquía.
- **Name:** El nombre del elemento o atributo. Los atributos están en cursiva.

Si el elemento no tiene ninguna anotación, es parte del nivel A del Diseño de Aprendizaje. Si tiene (*) es parte del Nivel B (B contiene a A). Si tiene (**) es parte del nivel C y si tiene (cp) el elemento es parte del IMS Content Packaging.

- **Explanation:** El significado y función del elemento.
- **Reqd:** Indica si el elemento o el atributo es obligatorio (mandatory) (M) u opcional (O).
- **Mult:** Indica la multiplicidad del elemento o atributo: 1 (el elemento ocurre una vez), 0..1 (cero o una vez), 0..n (puede aparecer 0 o más veces), 1..* (ocurre una o mas veces), - (multiplicidad indeterminada).
- **type:** Indica el tipo del elemento o atributo: Container (envuelve uno o más elementos), Choice (envuelve una selección de múltiples elementos), Sequence (envuelve una colección ordenada de elementos múltiples), Group (lugar para una jerarquía organizada), String (lugar para caracteres), Any (lugar para cualquier otra construcción), Empty (nota final que no contiene más caracteres).

Para distinguir entre referencias dentro del modelo y referencias a recursos en el paquete de contenido:

- El atributo "ref" se refiere a un elemento con identificador dentro del diseño de aprendizaje: <nombre ref="..."/>, se refiere a un elemento nombre dentro del diseño de aprendizaje.

El atributo "identifierref" se refiere a un recurso en el paquete de contenido: <item identifierref="..."/>

14.9.4.1. Nivel A del Modelo de Información

Modelo conceptual:

El modelo conceptual UML para el nivel A está en la siguiente figura:

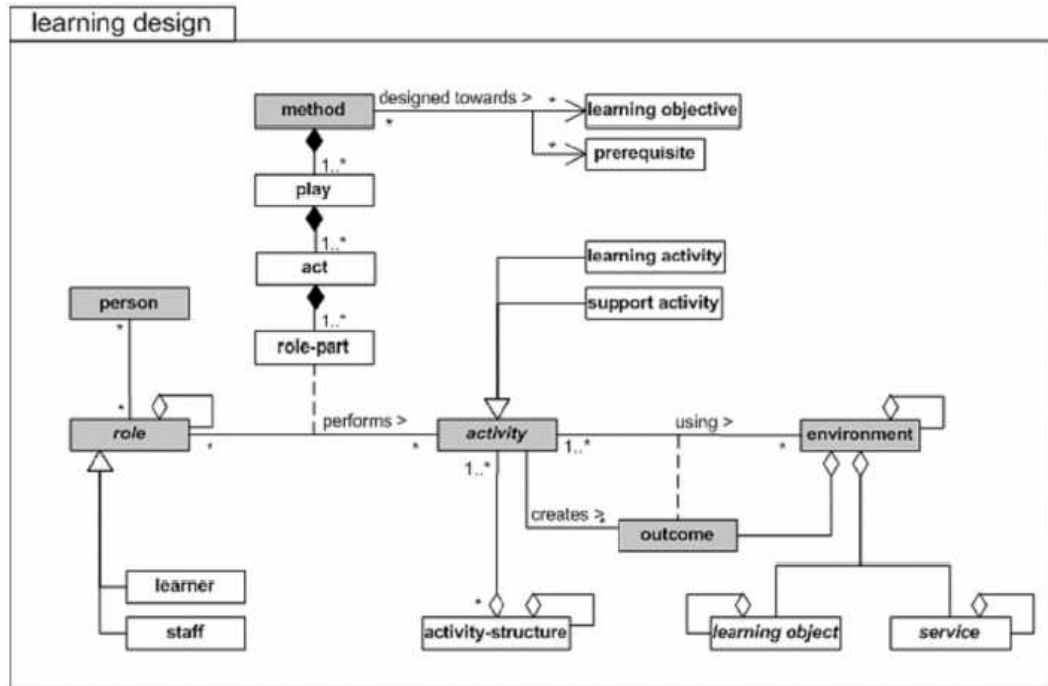
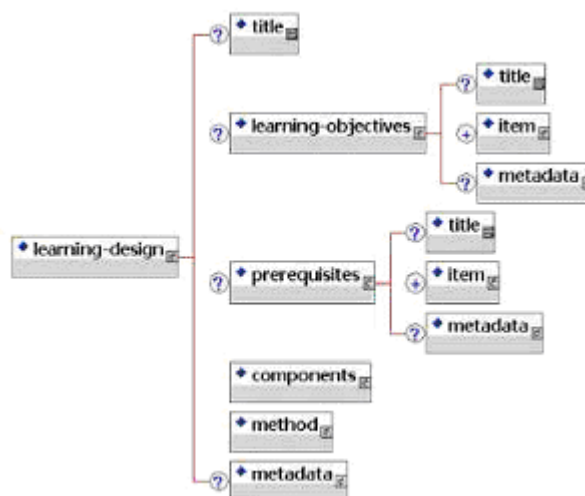


Figure 3.1 - Conceptual model of level A.

Tabla de Información "learning-design":





learning-design					
No.	Name	Explanation	Reqd	Mult	Type
0	learning-design	This element specifies the learning design.	-	-	Sequence
0.1	identifier	An identifier that is unique within the learning design file (ID).	M	1	ID
0.2	version	A version number.	O	1	String
0.3	url	Specifies a URL.	M	1	String
0.4	level	Specifies the lowest level of Learning Design that the document instance is valid against. The letter is specified with one of the following characters: A, B, C, a, b or c.	O	1	String
0.5	sequence-used	Boolean, when set to 'true' IMS Simple Sequencing is included at the appropriate places in the document instance. Default is false.	O	1	Token group
0.6	title (cp)	Similar to IMS CP. A short name given to the resource, suitable for rendering in user-agents.	O	0..1	String
0.6.1	e-dtype	See IMS CP.	O	1	Token
0.7	learning-objectives	Learning objectives describe the intended outcome for learners. Learning-objectives and prerequisites contain a standard organization of items, referring to resources or sub manifests. Resource types connected to learning objectives and prerequisites can be webcontent, imslcontent or it can point to an IMS Reusable Competency Definition Schema. There are two locations where learning-objectives and prerequisites are specified: - At the level of the learning design (in the root of learning-design) - At the level of learning-activities (within learning-activities). The first ones are a more general description; the second ones are more concrete. There are two types of learning-objectives: 1 human readable descriptions (the items point to text resources) 2 machine-readable specifications. These are addressed through the href attribute of the resources pointed at. The learning-objectives schemas could be user-defined or fixed by an organization. In the latter case, the texts of the learning objectives are referred to (through href).	O	0..1	Container
0.7.1	{itemmodel} (cp)	A schema group. Similar to IMS CP itemmodel.	M	1	Group
0.8	prerequisites	Prerequisites are the entry-requirements for students, e.g., the pre-knowledge needed. For the item formats see the description of the element 'learning-objectives'.	O	0..1	Container
0.8.1	{itemmodel} (cp)	See above	M	1	Group
0.9	components	Specifies the building blocks used in the method section.	M	1	Sequence
0.10	method	The method contains a sequence of elements for the definition of the dynamics of the learning process. It consists of one or more play(s) (which could be interpreted as the runscript for the unit of learning) and a statement for the completion of the unit of learning.	M	1	Sequence
0.11	metadata (cp)	Include IMS Meta-Data here, using its namespace.	O	0..1	Sequence

Nombre estándar para el Fichero Manifiesto de la Unidad de Aprendizaje:

Tiene un nombre y localización pre-definidos, se llama "imsmanifest.xml", debe estar en la raíz del fichero de intercambio del paquete o en cualquier otra imagen empaquetada como un CD-ROM. El nombre debe estar en minúsculas.

Namespace estándar para Elementos IMS Learning Design: para Elementos IMS Learning Design:

El namespace a usar para los elementos del esquema del diseño de aprendizaje (learning design) es: http://www.imsglobal.org/xsd/imsl_d_v1_p0



14.9.4.2. Nivel B del Modelo de Información

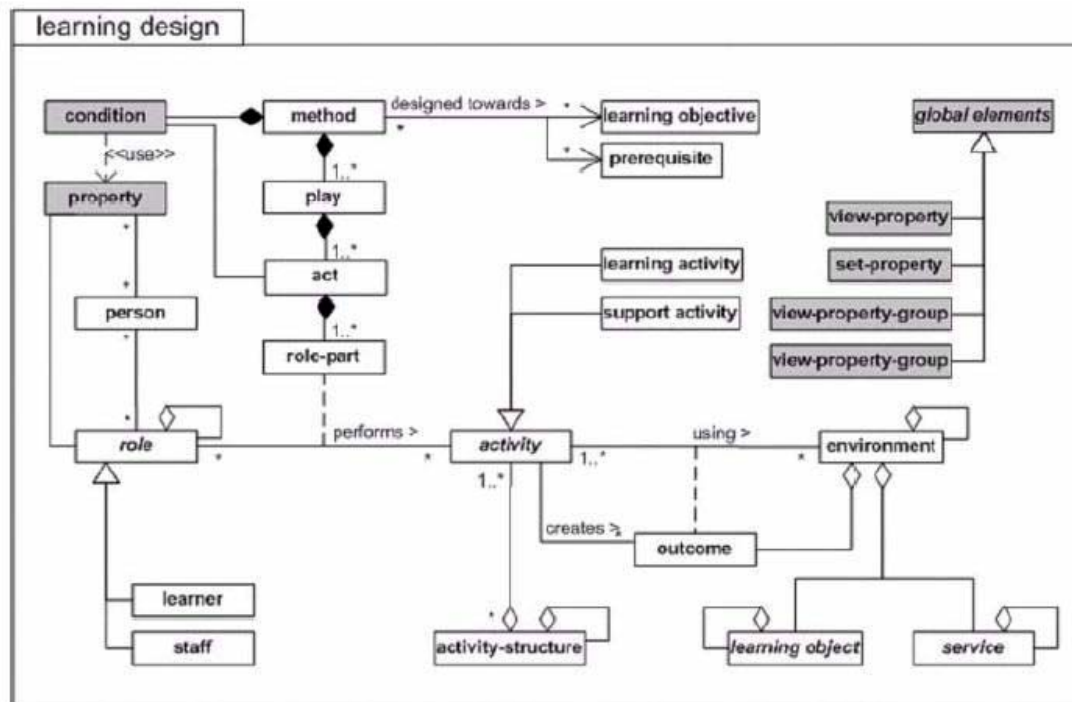
El nivel B proporciona elementos adicionales, estos son propiedades y condiciones, los cuales extienden significativamente la habilidad del diseñador de aprendizaje para controlar el flujo en una Unidad de Aprendizaje. Los principales elementos añadidos son Propiedades y condiciones.

La adición de propiedades y condiciones afecta a los siguientes modelos:

- El modelo de componentes (components) se extiende con el elemento properties, este es lugar donde se declaran las propiedades.
- Los modelos de complete-activity, complete-act, complete-play y complete-unit-of-learning se extiende para incluir el elemento when-property-value-is-set.
- El modelo on-completion se extiende para incluir el elemento change-property-value.
- El modelo de service se extiende para incluir el elemento monitor.
- El modelo de email-data se extiende con dos atributos (email-property-ref y username-property-ref):
- El modelo de time-limit se extiende con un atributo (property-ref) refiriéndose a una propiedad con datos.
- El elemento method se extiende para incluir el elemento when-condition-true.
- Se incluye un grupo separado de elementos-globales para leer y poner propiedades desde todos los tipos de esquemas de contenido basados en XML.
- Se usa el atributo global W3C clase para mostrar y ocultar condiciones de elementos de contenido en todos los tipos de esquemas de contenido basados en XML.

Modelo Conceptual:

El modelo conceptual UML para el nivel B está en la siguiente figura: (las clases en gris son añadidas al nivel A)



El sistema en tiempo de ejecución "user-agent" se espera que mantenga un registro de las "property-values" y las "property-definitions" para usuarios y roles, en lo que se llama el dosier.

Las propiedades se definen y/o declaran en learning-design/components/properties y se operan con los elementos de operación de propiedades "property-operation elements" como son: view-property, set-property, conditions, change-property...

Hay varios tipos de propiedades:

- **Las propiedades locales** (nombre del elemento: loc-property) se guardan con alcance local a la ejecución de una unidad de aprendizaje. Se definen y usan en la unidad de aprendizaje. El valor de esta propiedad es la misma para cada usuario en la ejecución de la unidad de aprendizaje, pero puede ser distinta en distintas ejecuciones.
- **Las propiedades globales** (nombre del elemento: glob-property) son accesibles fuera del contexto de una unidad de almacenamiento (por ej., por más de una unidad). Pueden ser definidas en una unidad de aprendizaje y usadas en otra. Se espera que las ejecuciones controlen si una propiedad global definida ya existe o no. Las propiedades globales - una vez definidas- no pueden cambiar nunca la definición. Así cuando la propiedad ya existe, la definición se ignora.

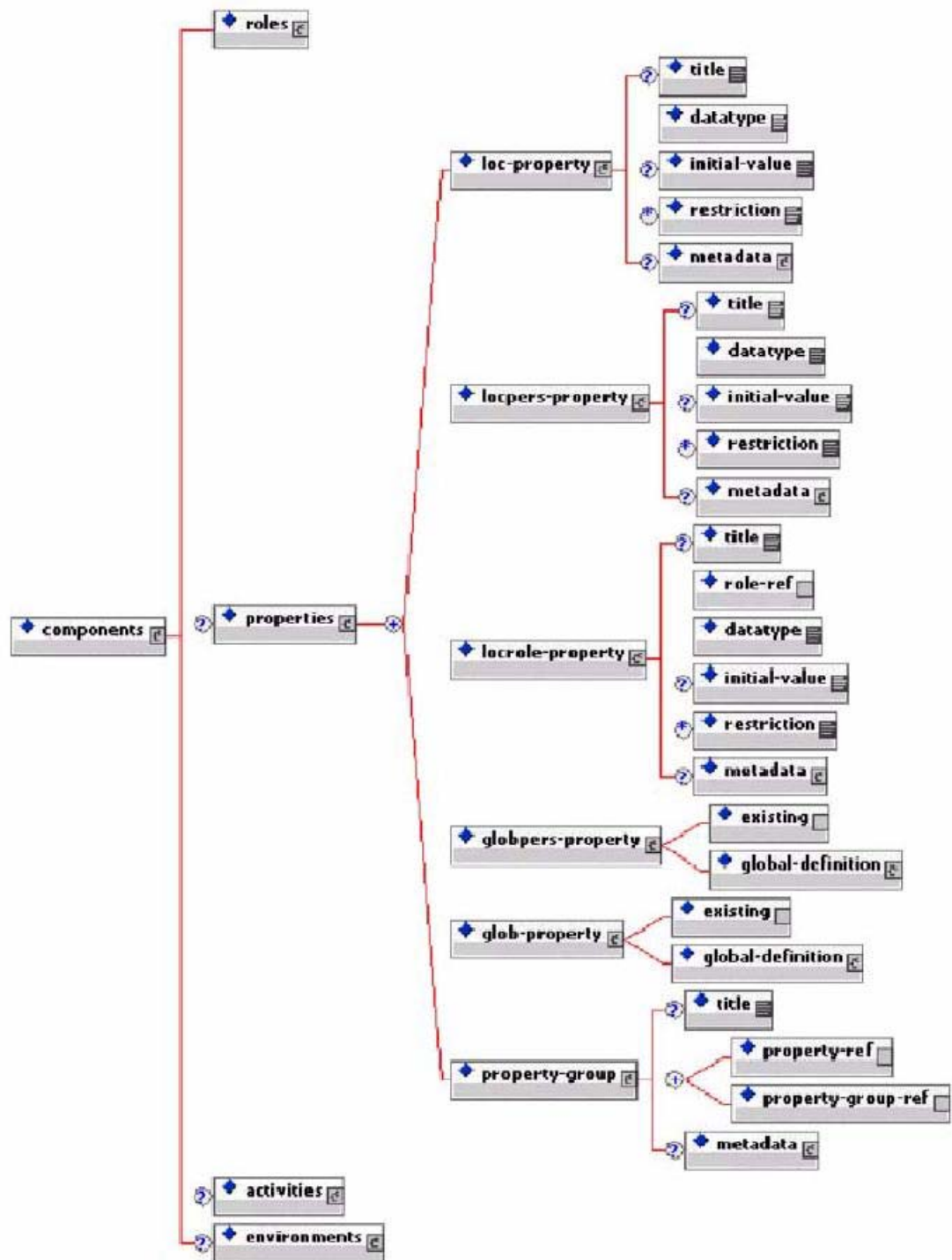


- **Las propiedades personales** (nombre de elem.: locpers-property y globpers-property) son pertenecidas por una persona (local o global). Se usan para personalización. Por ejemplo, un portafolio que trabaja sobre unidades de aprendizaje puede ser modelado con globpers properties (personal global). Las propiedades personales pueden ser almacenadas en un "dossier" personal y portátil.
- **Las propiedades de roles** (nombre del elemento: locrole-property) pertenecen a un rol y siempre son locales. Cada usuario en un rol específico puede acceder a esta propiedad y tiene el mismo valor en la misma ejecución de la unidad de aprendizaje.

Las propiedades globales tienen que ser mantenidas en un almacén persistente. La organización o institución que controla este almacén efectivamente, determina el alcance de las propiedades globales permitiendo o denegando el acceso al almacén. Puede haber un número de distintos sistemas en tiempo de ejecución accediendo al mismo almacén. El alcance de estas propiedades globales está por lo tanto extendida a todos estos sistemas.

Tabla de información "properties":

El elemento properties se añade al modelo de contenido del elemento components.





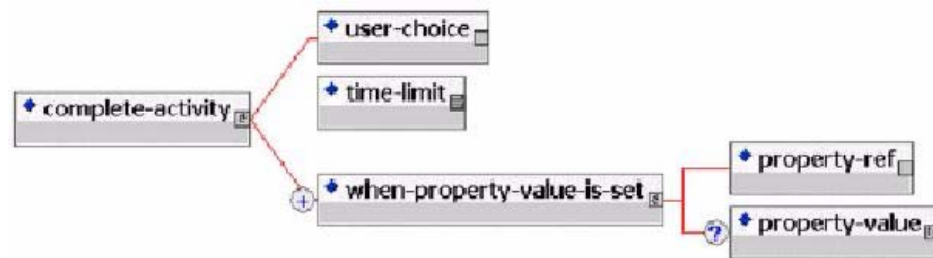
■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■

TABLA DE INFORMACIÓN "when-property-value-is-set":

El elemento when-property-value-is-set se añade a los modelos de contenido de los siguientes elementos del nivel A:

- Complete-activity.
- Complete-act.
- Complete-play.
- Complete-unit-of-learning.

En los cuatro se añade como el último elemento en el grupo. Ejemplo para complete-activity:



14.9.4.3. Nivel C del Modelo de Información

El nivel C añade la capacidad para especificar el envío de mensajes y poner nuevas actividades basadas en ciertos eventos. El sistema en tiempo de ejecución o el "useragent" se espera que soporten un mecanismo de notificación. Las notificaciones son mecanismos de eventos dirigidos, los cuales pueden ser dirigidos hacia elementos en el sistema o hacia usuarios humanos.

Las notificaciones afectan a los siguientes elementos de los modelos de contenido del nivel B:

- El modelo on-completion se extiende con un elemento notificación.
- El modelo then se extiende con un elemento notificación.
- Los elementos globales set-property y set-property-group son ambos extendidos con un elemento notificación.

Modelo Conceptual:

El modelo UML conceptual para el nivel C está en la siguiente figura. Las clases marcadas en gris son añadidas al modelo del nivel B.

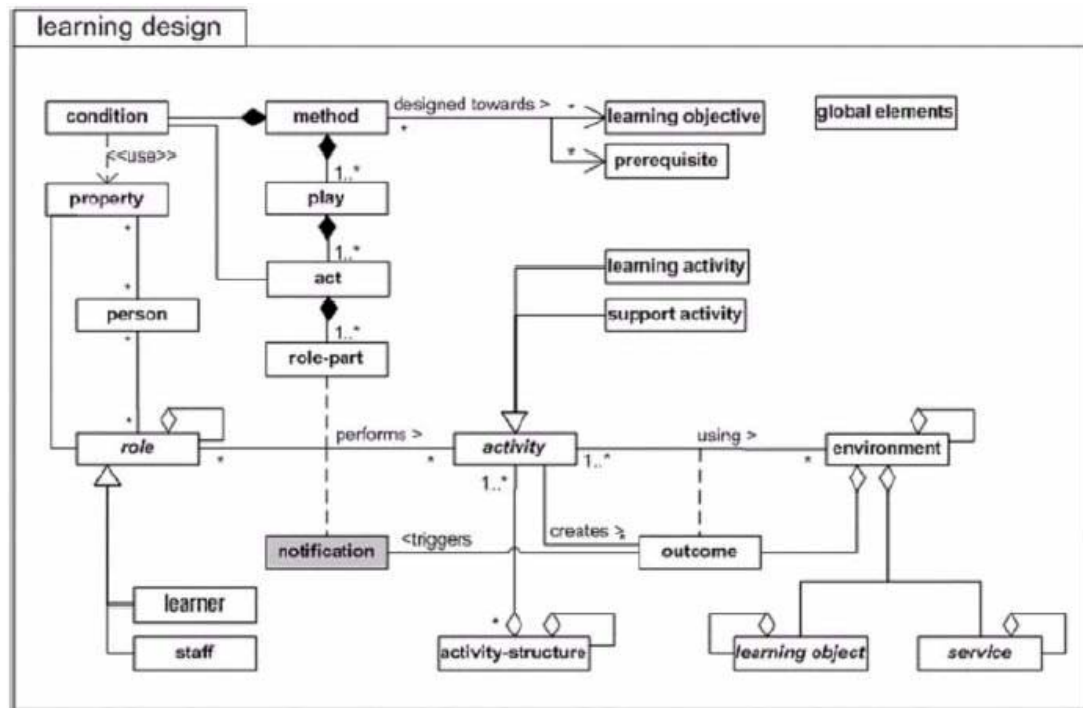
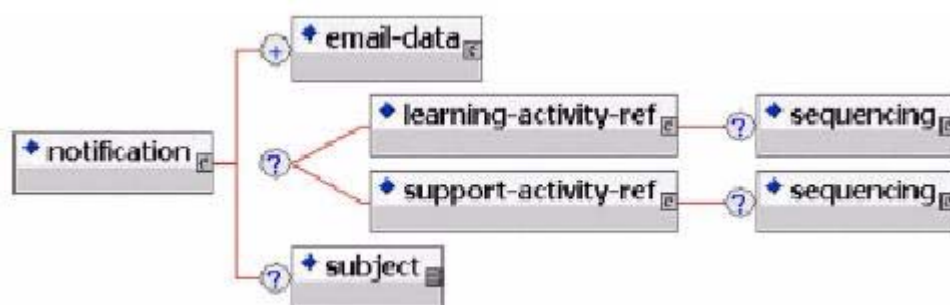


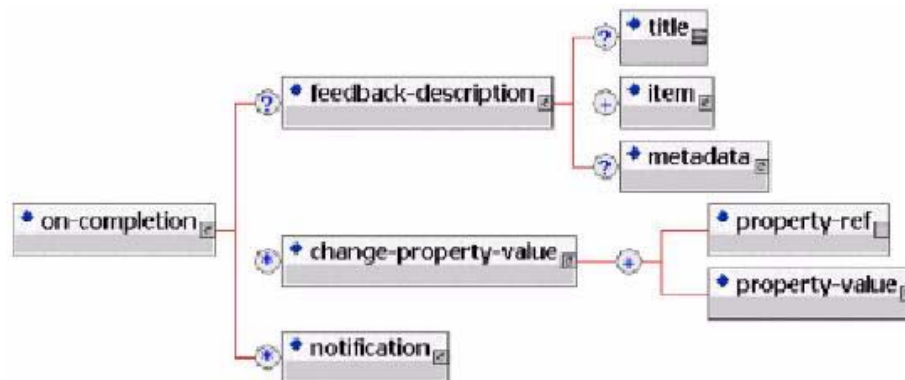
Tabla de información "notificación":



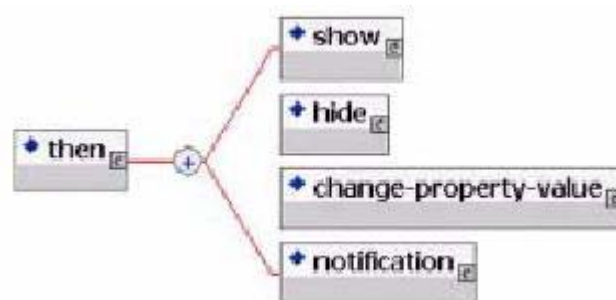


notification					
No.	Name	Explanation	Reqd	Mult	Type
0	notification (**)	A notification happens after an event that is known by the runtime environment. Such an event can be e.g., the completion of an activity, an expression evaluates to true, or a property value is set. The notification makes a new learning activity or a new support activity active for a role or it only sends a message. A notification is of the highest priority, meaning that an otherwise invisible item will be made visible and accessible to the user. Depending on the implementation an email message can be send to the user, notifying that a new activity has arrived (with a link to that activity in the message). The subject field can be set to a specific value (otherwise a standard message will be send). A notification can be inserted in external vocabularies (after an event like set-property), however, then the content must be provided in the package (because it contains references to identifiers in the package). When the identifier cannot be resolved the notification is ignored (but doesn't prevent the xhtml content from being presented).	-	-	Sequence
0.1	email-data	This is used for send-mail purposes (as a service in the environment, or in notifications). In level B, the properties on this element refer to the property resources where the relevant e-mail data can be found for the connected role. In level A, the source is not specified explicitly and is left to implementers to decide how to address the data needed. Both properties (email, username) should be available for all persons assigned to the role and also for the sending party.	M	1..*	Container
0.1.1	email-property-ref	Attribute contains a reference to the property containing the email address of the users being notified.	M	1	IDREF
0.1.2	username-property-ref	This attribute contains a reference to the property containing the user name of the users being notified.	O	1	IDREF
0.1.3	role-ref	Refers to the identifier of the resource of the role. The element can be used as an operand in an expression.	M	1	Empty
0.1.3.1	ref	Refers to an identifier within the learning design.	M	1	IDREF
0.2		Choice	O	0..1	Choice
0.2.1	learning-activity-ref	Refers to a learning-activity. The element can be used as an operand in a calculation or expression.	M	1	Container
0.2.1.1	ref	See above	M	1	IDREF
0.2.1.2	sequencing	This is where IMS Simple Sequencing Schema elements are namespaced into the Learning Design. It occurs only in activity-structures, on activity-lists and on single activities.	O	0..1	Sequence
0.2.1.2.1	schema (cp)	Similar to IMS Content Packaging (see IMS CP).	O	0..1	String
0.2.1.2.1.1	e-dtype	See IMS CP.	O	1	Token
0.2.1.2.2	schemaversion (cp)	Similar to IMS Content Packaging (see IMS CP).	O	0..1	String
0.2.1.2.2.1	e-dtype	See above	O	1	Token
0.2.2	support-activity-ref	Refers to a support-activity. The element can be used as an operand in a calculation or expression.	M	1	Container
0.2.2.1	ref	See above	M	1	IDREF
0.2.2.2	sequencing	See above	O	0..1	Sequence
0.3	subject (**)	It specifies the subject of a notification, to be presented to the notified actor when the notification is activated (e.g., in the mail-header (subject field).	O	0..1	String

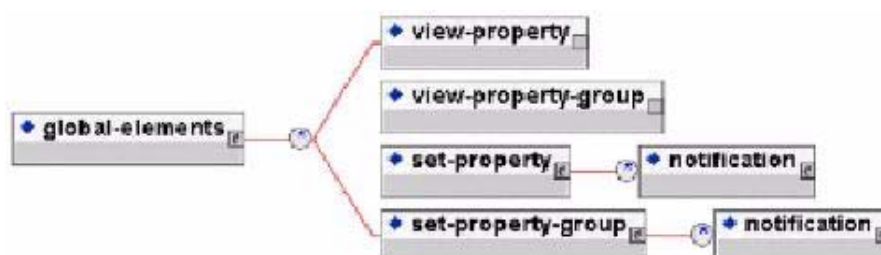
Extensión de "ON-COMPLETION": se añaden modificaciones al modelo on-completion:



Extensión de "THEN": se añaden modificaciones al modelo then: se añaden modificaciones al modelo then:



Extensión de "GLOBAL ELEMENTS": se añaden modificaciones los modelos global-elements: Set-property y Set-property-group:





14.9.5. Modelo de Comportamiento

Hay dos partes generales del modelo de comportamiento Learning Design:

1. Instanciación.
2. Tiempo de ejecución (runtime).

Instanciación:

Hay dos sub-partes principales, en adición a la instanciación de la unidad de aprendizaje en sí:

- **Instanciando Roles:** La parte principal de instanciar un learning design es la asociación de gente particular con los roles especificados en el diseño. Las Propiedades del Nivel B de Learning Design podrían tener que ser inicializadas para varios Roles.
- **Instanciando Servicios:** Una vez que se conocen los roles, los servicios genéricos pueden estar haciéndose pasar por o con miembros reales de acuerdo con los roles asociados con el uso del servicio, teniendo en cuenta los permisos acordados para cada rol.

Tiempo ejecución (runtime):

Los tres distintos niveles de Learning Design dan niveles incrementales de sofisticación para el comportamiento en tiempo de ejecución de una unidad de aprendizaje:

- **Nivel A del Diseño del Aprendizaje "Learning Design":**

Para el nivel A, el tiempo de ejecución también se divide en dos aspectos generales:

1. **Activity-structure Sequence and Selection:** Una estructura de Actividad "Activity-structure" puede contener como sub-elementos cualquier combinación de referencias a Actividad "Activity references", referencias a Estructuras de Actividad "Activity Structure references" y/o referencias externas a Unidades de Aprendizaje.

Para el nivel A del Diseño de Aprendizaje "Learning Design", un elemento estructura de actividad "activity- structure" tiene tres atributos que afectan al comportamiento secuencial:



- **Structure-type:** tiene un vocabulario fijo: {sequence, selection}. Determina si los sub-elementos van a ser entregados (delivered) como una secuencia o como una selección. Una selección es una estructura donde los usuarios pueden seleccionar y completar las actividades contenidas en cualquier orden. Como las activity-structures pueden estar anidadas, las selecciones pueden estar anidadas en otras secuencias o selecciones.
- **Number-to-select:** En un entero. Determina el número de sub-elementos que van a ser entregados (delivered) antes de que la selección structure-type sea considerada completa. Si el structure-type es una selección y el number-to-select es 1, entonces cualquiera de los sub-elementos va a ser entregado. Si el number-to-select no se define (o es igual al número de sub-elementos), entonces todos deben ser entregados pero en cualquier orden elegido por el usuario.
- **Sort:** Tiene un vocabulario fijo: {as-is, visibility-order}. El atributo "sort" determina el orden-de-ordenación en relación con la visibilidad. El orden por defecto en que las actividades se hacen visibles es en el orden especificado en la estructura activity-selection. Cuando vale "visibility-order", las actividades se presentan en el orden en que se harían visibles por las condiciones o notificaciones. Puede pensarse como imitando una bandeja de entrada donde las nuevas actividades se envían (hacen visibles) y por tanto disponibles el tiempo que los usuarios juegan los roles asociados. Esto sólo se soporta en los niveles B y C.

<sequence> element: En adición a, o mejor como una alternativa a, estos atributos, el elemento <sequence> de la especificación IMS Simple Sequencing estará opcionalmente disponible en la activity-structure y sus hijos. La presencia de un elemento <sequence> será tomada como sobreponiendo (overriding) cualquier valor de los atributos de arriba. El namespace de IMS Simple Sequencing tiene que ser usado en los elementos.

2. **Método "Method":** Un método tiene una o más escenas (plays). Las escenas son ejecutadas concurrentemente.

Una escena tiene uno o más actos. Los actos siempre se ejecutan en secuencia.

Un acto tiene uno o más role-parts. Estos están asociados con un y solo un rol, especificando la parte que este rol juega en el acto.



Un rol puede ser jugado por uno o más actores (p Ej., personas reales haciendo el rol de alumnos o personal de soporte). Los role-parts siempre se ejecutan concurrentemente.

Un role-part asocia un rol con una actividad o con un entorno, que contiene uno o más learning objects o services.

Complete-activity es igual a "activity completion rule". Especifica cuando una actividad está completada. En el nivel A la realización puede ser por elección-del-usuario, o alcanzando un límite-de-tiempo. En nivel B se extiende con when-property-value-is-set. On-completion especifica las acciones que van a ser ejecutadas en la realización de la actividad. En nivel A contiene sólo un elemento, feedback-description cuyo contenido de referencias será mostrado al usuario cuando completa la actividad. On-completion es extendido por el elemento change-property-value del nivel B, y por el elemento notification del nivel C.

- **Nivel B del Diseño de Aprendizaje:**

El nivel B incluye Propiedades y Condiciones. Todas las condiciones son precondiciones definidas en tiempo de desarrollo y deben ser evaluadas durante una ejecución:

1. Cuando se entra en la unidad de aprendizaje (nueva sesión).
2. Cada vez que se cambia el valor de una propiedad. Esto se aplica sólo a las siguientes situaciones:
 - 2.1. Propiedades que la persona ha accedido en el contexto de unidad de aprendizaje, y
 - 2.2. La propiedad tiene que ser evaluada en una de las expresiones en la unidad de aprendizaje.

Estas propiedades incluyen propiedades, que están disponibles en la expresión, pero se ponen automáticamente (p ej., time-unit-of-learning-started).

Se ejecuta (dispara) una acción de acuerdo con el éxito (true) o fallo (false) de la condición. La acción es mostrar u ocultar varios objetos, cambiar un valor de propiedad, o notificar a un rol.

El mostrar y ocultar acciones pone el atributo de visibilidad (isvisible) de diferentes objetos: actividades, entornos, ítems, escenas, activity-structures, unidades de aprendizaje y diferentes clases de objetos (puestos con el atributo "set").



Las propiedades pertenecen a los roles (propiedades local-role), las personas individuales en el rol (propiedades local-personal y global-personal), a la ejecución de una unidad de aprendizaje (propiedades locales) o son globales (propiedades globales). Hay cinco tipos distintos y pueden ser reunidos en property-groups.

Las propiedades pueden ser personales o no-personales: Las propiedades personales con claves para personalizar un learning design. Dado que muchos actores pueden jugar el mismo rol, las propiedades personales son asignadas al dossier individual (perfil o registro) de cada persona que juega ese rol. Las propiedades no-personales pueden ser propiedades globales fijadas (el mismo valor para todas las personas, independientemente del rol o unidad de aprendizaje) o propiedades locales que pertenecen a la ejecución de una unidad de aprendizaje y afectan a todos los usuarios, jugando ese rol o participando en la unidad de aprendizaje, respectivamente.

Las propiedades pueden ser también locales o globales: Las propiedades locales son creadas durante una ejecución particular de una unidad de aprendizaje y dejan de existir cuando la unidad termina. Las propiedades globales persisten más allá de la duración de la ejecución de una unidad de aprendizaje. También pueden ser usadas para acceder a valores de la información persistente del alumno.

Los cinco tipos diferentes de propiedades son: Local property (igual valor en cada ejecución para cada usuario), Local personal property (pertenece al usuario en el contexto de una ejecución por lo que podrá tener un valor distinto para cada usuario), Local role property (pertenece al rol en la ejecución), Global personal property, Global property.

- **Nivel C del Diseño de Aprendizaje:**

El nivel C añade notificación. Después de un evento se produce una notificación. Ejemplos de eventos que pueden disparar una notificación incluyen completar una actividad, una expresión que evalúa a cierto, o se establece el valor de una propiedad.

Una notificación en tiempo de ejecución puede poner la visibilidad de una actividad, que es activada entonces para un rol. Si una notificación para un rol particular pone la propiedad invisible de una actividad dada a cierto, esta actividad se hace disponible inmediatamente al rol, sin considerar ninguna otra configuración en el acto, secuencia o condición. Una notificación tiene la máxima prioridad.



El sistema en tiempo de ejecución envía mensajes de notificación a los roles (afectando a todos los actores jugando el rol). Las notificaciones se disparan cuando se cumplen ciertas condiciones o cuando un actor, en un rol con los permisos apropiados, las envía. En tiempo de ejecución, dependiendo de la implementación, es posible seleccionar un actor particular como el receptor de una notificación.

Un sistema en tiempo de ejecución podría guardar quien origina la notificación y hacerla visible al receptor.

Dependiendo de la implementación, un mensaje de email puede ser enviado al usuario, notificándole que una actividad nueva ha llegado (con un enlace a esa actividad contenida en el mensaje).

Una notificación puede ser insertada en vocabularios externos. Sin embargo, el contenido debe ser entonces provisto en el paquete (porque contiene referencias a identificadores en el paquete). Cuando el identificador no puede resolverse, la notificación se ignora.

Jerarquía de control:

Hay varias estructuras que tienen influencia sobre la visibilidad de actividades de aprendizaje u otras entidades. Para resolver posibles conflictos, hay una jerarquía de control:

Notify (Nivel C LD) una notificación en tiempo de ejecución puede poner la visibilidad de una actividad cuando es activada entonces.

Acts (Nivel A LD) determina si, cuando, y para qué roles una actividad, estructura de recurso o ítem va a ser usado.

Sequence (Nivel A LD) es un tipo de estructura-de-actividad y pone el orden para completar las actividades en la secuencia. Esto resetea los valores del atributo invisible, sin considerar las condiciones y los valores iniciales del atributo invisible.

Condition (Nivel B LD) podría resetear la propiedad invisible de una actividad, estructura de recurso o ítem, sin considerar sus valores actuales.

Isvisible (Nivel A LD) este atributo dice si una actividad, estructura de recurso o ítem es mostrada al alumno.

Esto significa que los valores "invisible" pueden ser sobrescritos por las condiciones... y que las notificaciones son el mecanismo más fuerte.



Un método puede ser expresado como una tabla:

Method			
Play 1	Act 1	Role 1	Activity 1
		Role 2	Activity 2
		Role 3	Activity 3
	complete act requirements		
	Act 2	Role 1	Activity 5
		Role 4	Activity 6
	complete act requirements		
complete play requirements			
Play 2	Act 3	Role 1	Activity 9
		Role 3	Activity 10
		Role 4	Activity 11
	complete act requirements		
	Act 4	Role 1	Activity 3
		Role 2	Activity 1
		Role 3	Activity 2
complete act requirements			
complete play requirements			
complete method (unit of learning) requirements			

Por ejemplo: una escena puede especificar lo siguiente:

Method			
Play 1	Act	Role	Role-part assigned
	1.1	Teacher	support-activity: teacher-introduction
	1.2	Student	learning activity: introduction
	complete act when all individual students are finished		
	2.1	Student	activity-structure: lessons&discussions
	2.2	Teacher	activity-structure:teaching
	complete act when all teachers complete activity		
	3.1	Student	learning-activity: assessment
	3.2	Teacher	support-activity: closing-activities
	complete act when teacher has completed		
complete play when last act has been completed			
complete method when play 1 completed			



Esto se representa en un Learning Design Method como sigue:

```
<method>
  <play id="play1">
    <act id="act1">
      <role-part id="part11"><role-ref ref="Teacher"/><support-activity-ref
ref="teacher-introduction"/></role-part>
      <role-part id="part12"><role-ref ref="Student"/><learning-activity-ref
ref="introduction"/></role-part>
      <complete-act><when-role-part-completed ref="part11"/></complete-act>
    </act>
    <act id="act2">
      <role-part id="part21"><role-ref ref="Student"/><activity-structure-ref
ref="lessons&discussions"/></role-part>
      <role-part id="part22"><role-ref ref="Teacher"/><activity-structure-ref
ref="teaching"/></role-part>
      <complete-act><when-role-part-completed ref="part22"/></complete-act>
    </act>
    <act id="act3">
      <role-part id="part31"><role-ref ref="Student"/><learning-activity-ref
ref="assessment"/></role-part>
      <role-part id="part32"><role-ref ref="Teacher"/><support-activity-ref
ref="closing-activities"/></role-part>
      <complete-act><when-role-part-completed ref="part32"/></complete-act>
    </act>
    <complete-play><when-last-act-completed/></complete-play>
  </play>
  <complete-unit-of-learning><when-play-completed ref="play1"/> </complete-unit-of-learning>
</method>
```

Nota: id en el ejemplo es la abreviatura de "identifier". Los nombre específicos de los identificadores son arbitrarios.

Traducido en texto, esto significa:

Cuando entramos en la unidad de aprendizaje, la escena empieza con el acto número 1. Todas las personas asignadas al rol "Teacher" obtienen la actividad-de-soporte "teacherintroduction". Al mismo tiempo todas las personas asignadas al rol "Student" obtienen una actividad de aprendizaje llamada "introduction". El acto número 1 se completa cuando todas las personas del rol Estudiante han completado la introducción.

Entonces empieza el acto 2 asignando una estructura-de-actividad llamada "lessons & discussions" para las personas del rol Estudiante y asignando a la vez la estructura de actividad "teaching" a los Profesores. Este acto se cierra cuando el profesor completa el acto (asumiendo que hay una restricción de una persona en el rol profesor).

Entonces el acto 3 empieza. Etcétera. La escena se cierra cuando el último acto ha sido completado.

En un método se puede especificar más de una escena. Estas escenas corren en paralelo, independientes unas de las otras. Esto es necesario cuando modelamos diseños más complejos.

En vez de una tabla, se puede usar un diagrama de actividad UML o un Mapa de Gantt o (relativo) un horario. Diagrama de actividad UML, usando un swimlane para cada rol:

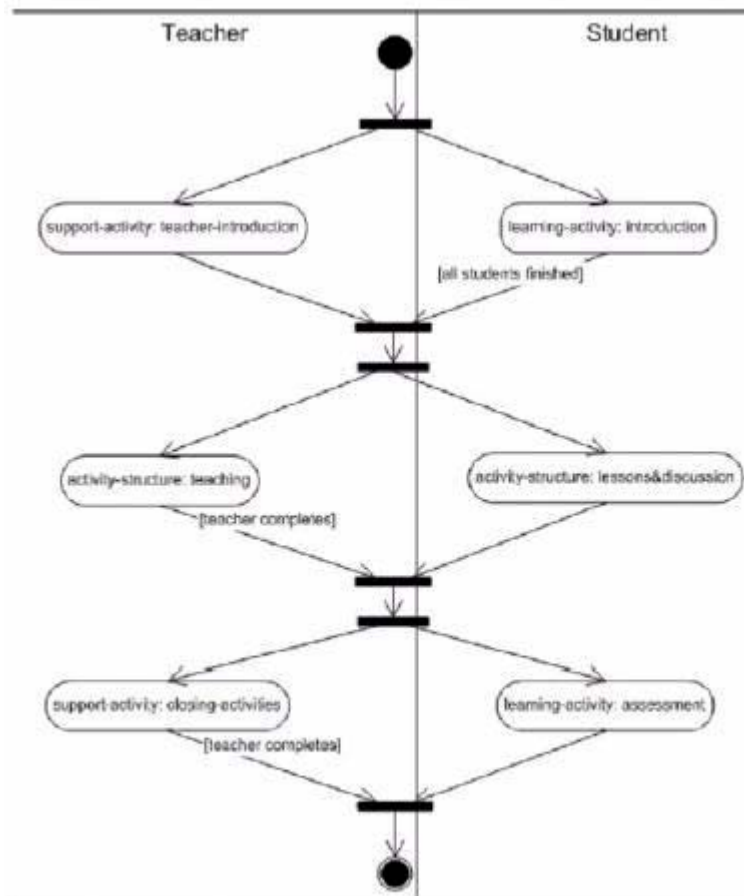


Figure 4.1 - UML activity diagram, using a swimlane for each role.

En la siguiente figura se ilustra un Mapa de Gantt:

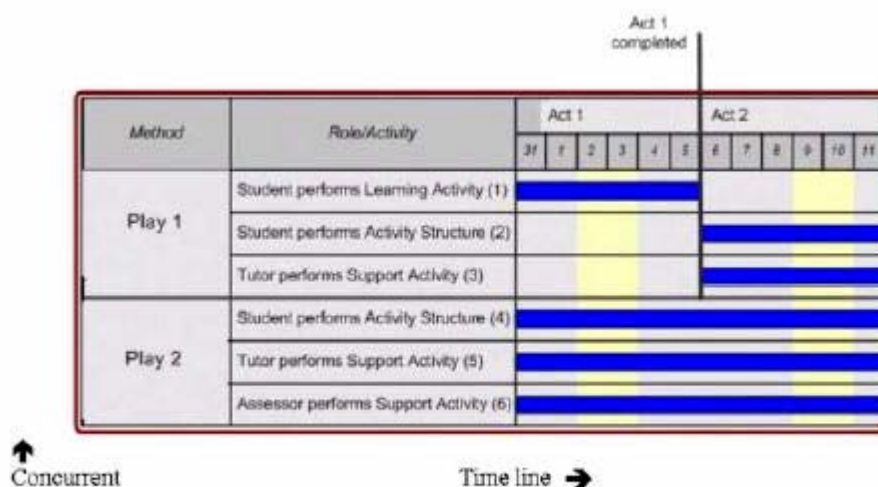


Figure 4.2 - An illustration of a Gantt Chart.

14.9.6. Modelo de Comportamiento (y II)

Hay otros conceptos que hay que aclarar dentro del Modelo de Comportamiento:

- **Instanciando una Unidad de Aprendizaje:** Se necesitan un número de pasos para traducir un documento XML IMS Learning Design en un "curso" vivo con el que un estudiante (y un profesor) interactúa a través de, por ejemplo, un navegador. Es necesario especificar qué medio se va a usar para publicar la unidad (típicamente la Web), qué elementos de media específicos hay (video, audio, dibujos, etc.), qué lenguaje será usado, quién participará y en qué roles, la fecha de comienzo y finalización para la ejecución.
- **Establecimiento por Instanciamiento de Roles y Servicios:** Se necesita dar información sobre la gente que jugará los roles de alumnos y profesorado.

Para el Nivel B de IMS Learning Design, cuando asignamos una persona a un rol, sus propiedades necesitan estar en su dossier. Cualesquiera propiedades personales locales atribuidas por el rol a cada persona, necesitan establecerse y poner sus valores por defecto, si los hay. Las propiedades de rol, que se aplican a todos los miembros que juegan ese Rol, también necesitan ser establecidas, como las propiedades que pertenecen a la unidad de aprendizaje, que se aplican a todas las personas de cualquier rol en la unidad de aprendizaje. También hay propiedades globales, que persisten a través de todas las unidades y usuarios, que tienen que ser localizadas e instanciadas si no se ha hecho ya.

Una vez que se conocen los participantes y sus roles, los servicios, que dependen de esta información, se pueden establecer.



- **Proceso de Activación:** Cuando activamos una unidad de aprendizaje, el elemento method tiene que ser localizado en la unidad. Siempre debe haber uno y solo uno. Este y sus sub-elementos controlan el comportamiento de la unidad de aprendizaje como un todo. Esto crea un "flujo de aprendizaje".

Funcionalmente, un método está hecho de uno o más elementos escénicos (play elements). Estos son funcionalmente independientes y corren en paralelo, así cada elemento escénico tiene que ser instanciado cuando la unidad de aprendizaje está inicializada primero.

El Método, Escenas, Actos y Role-parts están anidados unos dentro de otros, dando tres niveles dentro de un Método.

En el nivel más alto, el método consiste en 2 elementos, escena y complete-unit-oflearning.

En la inicialización, todos los elementos escénicos se activan a los miembros de los roles que participan en él.

Los elementos escénicos se desenrollan en unas series de uno o más actos, que siempre se ejecutan en secuencia. Ningún acto se hace visible al/los rol/es que participan en él hasta que el acto previo se complete. Este factor puede usarse para sincronizar las actividades de estos roles.

Las actividades pueden tener un secuenciamiento complejo, así, es posible tener una escena de "un-acto" que tiene internamente una secuencia compleja.

Una escena tiene un identificador y una propiedad invisible. También tiene un título y meta-datos. El acto o actos que constituyen el cuerpo principal de la escena, y la escena completa especifican las condiciones de completitud y las acciones opcionales que se tomarán cuando la escena esté completa.

Un acto tiene uno o más role-parts. Estos siempre corren en paralelo.

Cada role-part asocia exactamente un rol con exactamente un tipo de actividad (incluida la ejecución de otra unidad de aprendizaje y estructuras-de-actividad), o con un ambiente (equivalente a una organización en el Paquete de Contenido).

El mismo rol puede ser asociado con diferentes actividades o entornos en diferentes role-parts, y la misma actividad o entorno puede ser asociado con diferentes roles en diferentes role-parts. Sin embargo, el mismo rol sólo puede ser referenciado una vez en el mismo acto. Si múltiples actividades o entornos necesitan ser asociados para el mismo rol, debe usarse una estructura-de-actividad o un entorno envoltura.



Cuando se activa un acto en una escena, todos los role-parts en el acto van "a-escena" o se vuelven vivos. Dependiendo de la implementación, los jugadores de los roles referenciados por las role-parts podrían entonces tener sus actividades asociadas (o entornos) visibles en su "árbol-de-actividad" y cualquier contenido asociado con la actividad visible. Sin embargo, si el atributo invisible de una actividad o ítem es "falso", el enlace en el árbol-de-actividad podría hacerse visible, pero el contenido no ser accesible.

- **Reglas de Completitud:** A cualquier nivel en un método es posible especificar las reglas cuando una role-part, escena o unidad de aprendizaje es completada.

En el nivel más bajo, las role-parts tienen que completarse. La actividad que una role-part referencia puede ser una actividad-de-aprendizaje, una actividad-de-soporte, una estructura-de-actividad o una (sub) unidad-de-aprendizaje. Se completan cuando las condiciones complete-activity se cumplen (para actividades de aprendizaje / soporte), o cuando una estructura-de-actividad o una unidad de aprendizaje referenciada se completa.

Una estructura-de-actividad con tipo secuencia se completa cuando la última entidad referenciada se completa. Una estructura-de-actividad con tipo selección se completa cuando todas las entidades referenciadas contenidas se completan o cuando el número de entidades en number-to-select se completan. La completitud de una estructura-de-actividad o (sub)unidad-de aprendizaje está determinada así por la completitud de actividades de soporte y de aprendizaje "atómica" contenidas en ella.

La completitud de una actividad de soporte o de aprendizaje "atómica" está determinada por elección del usuario o cuando un límite de tiempo se alcanza (nivel A). Cuando no hay una regla de completitud explícita, la completitud se pone a ilimitada, significando que siempre es completada. En el Nivel A Learning Design, los próximos tres niveles, acto, escena y unidad de aprendizaje, cada uno tiene tres opciones de completitud.

En el nivel siguiente, un acto se completa cuando uno o más role-parts referenciados se completan, o por elección del usuario o cuando un límite de tiempo se alcanza. En el nivel siguiente otra vez, una escena se completa cuando el último (final) acto se completa, o por elección del usuario o cuando un límite de tiempo se alcanza. Finalmente, en el nivel más alto, la unidad de aprendizaje se completa cuando una o más escenas referenciadas se completan, o por elección del usuario o cuando un límite de tiempo se alcanza.

Hay que fijarse que la completitud de un elemento de nivel-superior, por elección del usuario o límite de tiempo, termina con todos los componentes del nivel menor contenidos en él.



El Nivel B de Learning Design añade la opción when-property-value-is-set como una adición a las tres opciones del Nivel A. Este elemento contiene una property-reference y un property-value. Cuando la propiedad referenciada tiene el valor especificado en el property-value, la condición evalúa a cierto y el acto, escena o unidad se completa.

Las condiciones de completitud para el acto, escena y unidad de aprendizaje son todas las mismas, excepto la primera condición que es única para cada una.

El valor de la propiedad puede ser especificado como un valor literal, un valor calculado, o como el valor de otra propiedad.

- **Al Completarse (On Completion):** Cuando una actividad de aprendizaje, actividad de soporte, acto, escena o unidad de aprendizaje se completa, se ejecutarán las acciones contenidas en el elemento oncompletion. Las opciones on-completion son las mismas para una actividad de aprendizaje, actividad de soporte, acto, escena y una unidad de aprendizaje. Se considera que una role-part está completa cuando sus actividades relacionadas están completas y no tiene elemento on-completion propio. Igualmente con estructuras de actividad...

En el Nivel A Learning Design, la única tarea que puede especificarse para ser llevada al completarse es la de proveer retroalimentación al usuario con el elemento descripción-de-retroalimentación (feedback-description). Esto apunta a un recurso donde puede encontrarse la descripción. Tras completar una actividad esta página Web será mostrada al usuario.

Nivel B Learning Design añade la opción de cambiar uno o más valores de las propiedades con el elemento change-property-value. Este elemento (también) contiene una property-reference y un property-value. Esto especifica que al completarse, la propiedad referenciada será puesta al valor especificado en el property-value.

Nivel C Learning Design añade la opción de enviar una o más notificaciones. (Ver la Vista Previa del Modelo de Comportamiento para una descripción de las Notificaciones.



- **Grabando Resultados y su Mapeado en IMS LIP:** Los resultados de las actividades de cada alumno en la Unidad de Aprendizaje típicamente serán guardados y mantenidos. El learning design presupone alguna forma de "dossier" o registro del alumno que se usa para tener y mantener las varias propiedades personales que son parte del Nivel B Learning Design. Las propiedades personales locales necesitan ser mantenidas sólo durante la duración de la unidad de aprendizaje, aunque podrían durar múltiples sesiones. Las propiedades personales globales se intenta que persistan indefinidamente y podrían formar parte de un registro permanente del alumno.

El learning design no especifica cómo deberían ser guardadas.

Sin embargo, no importa que las propiedades IMS Learning Design y grupos de propiedades se mapeen directamente en los elementos Actividades y Evaluaciones de las especificaciones del IMS Learner Information (LIP). Esto significa que podría necesitarse que la especificación IMS LIP pueda usarse para transportar los resultados generados por los alumnos en su uso de unidades de aprendizaje entre distintos sistemas.

14.9.7. Extensibilidad

Para permitir a los desarrolladores la máxima flexibilidad posible, las obligaciones XML de un Learning Design podrían extenderse con extensiones personalizadas. Todos los elementos que sirven como contenedores de otros elementos, p Ej., choice groups, podrían extenderse para incluir nuevos elementos. Un ejemplo es la extensión del lenguaje de expresiones con elementos como average, random o median. Los elementos que contienen tipos de datos (p Ej., string, integer) y elementos con un modelo de datos "cerrado" no deben extenderse. Las extensiones deben dar referencias (p Ej., vía namespacing) a la fuente de las extensiones. Una buena convención es empezar el nombre de cada elemento extendido con una "x-", p Ej., un posible elemento nuevo sería "_x-random".

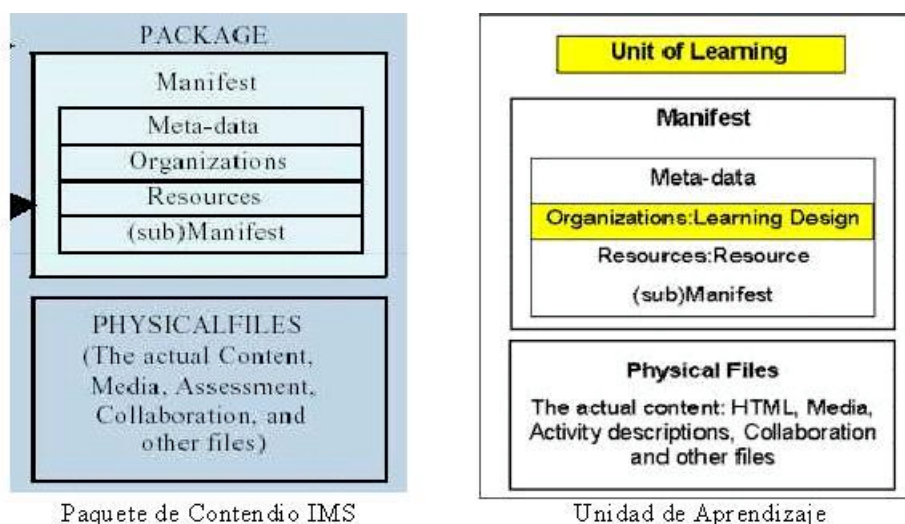
El modelo de información ya indica instancias que pueden ser sustituidas y/o extendidas con otros esquemas.



El mecanismo de extensión es flexible, pero las instancias de los documentos resultantes no son válidas más a los esquemas de IMS LD y causan problemas con la interoperabilidad y re-uso. Hay al menos dos casos donde las extensiones pueden causar problemas a los desarrolladores. El primer caso es cuando se requiere la interoperabilidad con otras herramientas o vendedores del paquete de contenido. Las extensiones personalizadas deben estar de acuerdo entre las partes individuales haciendo muy difícil la interoperabilidad global. El segundo caso es cuando un desarrollador desea añadir extensiones y también da o altera un esquema que permitirá la validación del documento. Cada esquema (DTD, XDR, o XSD) requiere una aproximación diferente para manejar extensiones que puedan ser validadas.

14.9.8. Comparación del "IMS Content Packaging" con otros estándares IMS

- Este estándar es muy parecido en cuanto a objetivos al **IMS Content Packaging Specification (Paquete de Contenido IMS)** que también describe estructuras de datos que se utilizan para desarrollar un proceso enseñanza- aprendizaje en Internet y así permitir el intercambio de información entre aplicaciones que usen el mismo estándar. Pero hay muchas diferencias estructurales entre ambos que hacen que no sean compatibles entre sí, ya que, el principal uso del Diseño de Aprendizaje es crear o modelar Unidades de aprendizaje, que no son más que Paquetes de Contenido modificados a los que se añade un elemento nuevo *<learning design>* dentro del elemento *<organizations>* que se encuentra en el Manifiesto y que por lo tanto, pasan de ser un **Paquete de Contenido IMS** a una **Unidad de Aprendizaje**, como se ve en la siguiente figura:





A parte de este cambio, habría que quitar todas las demás organizaciones que se encontraran en el elemento *<organizations>* y colocarlas en sub-manifiestos, sino, si las dejáramos dentro del elemento *<organizations>* junto al nuevo elemento incorporado *<learning design>*, serían ignoradas. Estos cambios se ven en el cuadro comparativo expuesto más abajo.

- Las especificaciones (estándares) IMS hasta esta última, reflejan un modelo para un sólo usuario, un estudiante, que interactúa con los contenidos para ir aprendiendo y que puede ser evaluado de esos conocimientos que va adquiriendo. **El Learning Design (Diseño de Aprendizaje) nivel A** da la capacidad de diseñar Unidades de Aprendizaje que incluyen simultáneamente diversos roles o papeles, cada uno de los cuales puede ser interpretado por varios actores. Esto permite que las Actividades correspondientes sean especificadas en "flujos de aprendizaje" coordinados que son análogos para flujos con objetos del mismo tipo o grupo. De esta manera, esta especificación soporta aprendizaje en grupo y de colaboración de muy diferentes tipos, cuya importancia está cada vez más reconocida en las esferas educacionales y en la formación comercial.

Esta especificación también puede ser usada para soportar un modelo de un solo estudiante mediante la creación de una Unidad de Aprendizaje con un único role definido y ninguna interacción entre los estudiantes. Además de que el Diseño de Aprendizaje (Learning Design) permite una mayor reutilización de los componentes ya que se separan las Actividades, de las **Estructuras de Actividades** (grupos de actividades que aportan la ventaja de hacer que las diferentes actividades en grupos puedan ser opcionales y no siempre secuenciales como lo eran antes, ver cuadro comparativo más abajo: *<activity-structure identifier=" " number-to-select="2" structure-type=" selection sequence">* reúne dos actividades y permite seleccionar una de ellas), de los recursos y de los roles y que luego se reúnen en un **Método** que se encarga de interrelacionarlas todas entre sí (es como una obra con diferentes actores en cada acto).

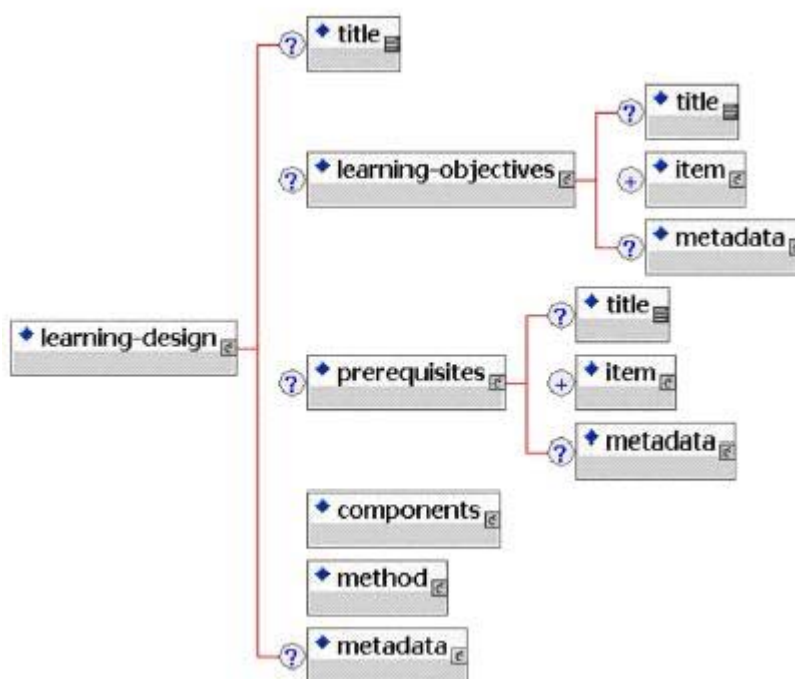
El nivel B añade una mayor personalización de los estudiantes, soportando conocimiento anterior al curso, accesibilidad, preferencias... Facilitando así, por ejemplo, la definición de entornos de trabajo, objetivos y prerequisites que permitirían una mayor individualización en las tareas de cada alumno.

El nivel C añade la capacidad de envío de mensajes entre los diferentes componentes del sistema y entre roles. Esto añade una nueva dimensión ya que permite diferentes flujos de educación conducidos por eventos y en tiempo real.

- En la siguiente tabla se muestran algunas de las modificaciones que se realizan en el *imsmanifest.xml* al usar el estándar del que hablamos: [Ver tabla](#)

14.9.9. Relación del estándar con los Meta Datos

- Como ya se ha mencionado en otros estándares, se pueden usar los meta-datos para referenciar un (Learning Object) **un Objeto de Aprendizaje** cualquiera del estándar, que es cualquier entidad digital o no, que pueda ser usada, rehusada o referenciada durante el aprendizaje (con tecnologías), por ejemplo, material impreso, temas a estudiar, ejercicios, textos, cursos, programas para estudiar e incluso personas. En general suelen ser elementos más pequeños que los cursos y que pueden ser rehusados en diferentes cursos, que en su mayoría no pueden ser rehusados ni intercambiados. Estos Objetos de Aprendizaje pueden ser entidades a las que se accede con Meta-Datos, que se guardan de forma separada a los Objetos de Aprendizaje a los que se refieren, por ejemplo en Bases de Datos. Una imagen de un ejemplo de como se integran los Meta-Data en este estándar, aparece en la siguiente figura:



Donde cada vez que aparece el recuadro metadata, significa que ahí se incluyen los Meta-Data IMS correspondientes siguiendo el estándar y mediante el namespace correspondiente. El elemento `<metadata>` es opcional y puede aparecer cero, una o más veces como se explica a continuación.

Dentro del estándar IMS Learning Design se proporciona un mecanismo para la inclusión de meta datos dentro de los diferentes elementos del paquete. Así, es posible añadir meta-datos para describir a los diferentes elementos que forman parte de una unidad de aprendizaje: manifiesto, objetivos de aprendizaje, prerequisites, componentes, métodos.



Al igual que en el IMS Content Packaging el meta-dato siempre describe al elemento dentro del cual está incluido. Así, el meta-dato del manifiesto describe la Unidad de Aprendizaje como un todo, y así el de cada elemento del árbol donde aparece un elemento *<metadata>* (la figura está incompleta, el árbol de elementos es mucho mayor). Hay que tener en cuenta que si el elemento tiene (sub)elementos, el meta dato describirá al elemento y los (sub)elementos como un todo.

- **Estructura de los meta datos:**

En el estándar IMS Learning Design hay dos elementos *<metadata>* diferenciados: el del manifiesto y el de los demás (sub)elementos al igual que ocurría en el Content Packaging:

- ***<metadata>* dentro de *<manifest>*** Aparece cero o una vez dentro del elemento *<manifest>*. Los elementos son:
 - *<schema>* Describe el esquema usado (por ejemplo: IMS Content, ADL Content, etc.). Este elemento aparece cero o una vez. No tiene atributos ni elementos. Ejemplo: *<schema>IMS Content </schema>*
 - *<schemaversion>* Dice la versión del esquema anterior. Si no aparece la versión, se asume que es "1.1". Este elemento aparece cero o una vez. No tiene atributos ni elementos. Ejemplo: *<schemaversion>1.1</schemaversion>*
 - *IMS Meta-Data* Se puede añadir cualquier elemento definido en la especificación IMS Meta-Data. El número de veces que puede aparecer este elemento depende del estándar IMS Meta-Data.
- ***<metadata>* dentro del resto de elementos:** Aparece cero o una vez dentro del elemento que lo contenga. Se puede añadir cualquier elemento definido en la especificación IMS Meta-Data. El número de veces que puede aparecer este elemento depende del estándar IMS Meta-Data.
- **Conclusión** El elemento *<metadata>* se define según la especificación IMS Meta-Data y únicamente en el nivel del *<manifest>* se incluye el esquema usado y su versión (en los demás se supone que se aplica el mismo esquema definido en el manifiesto).

No se define ningún modo estándar de uso, es decir, la implementación es libre: puedes usar ficheros externos, inclusión de namespace (*<imsmd:nombreelemento>*),...



14.9.10. Ver evaluación en Internet

14.10. IMS LEARNER INFORMATION PACKAGE SPECIFICATION

14.10.1. Introducción

El paquete de información del estudiante esta basado en un modelo de datos que describe las características que un estudiante necesita para...

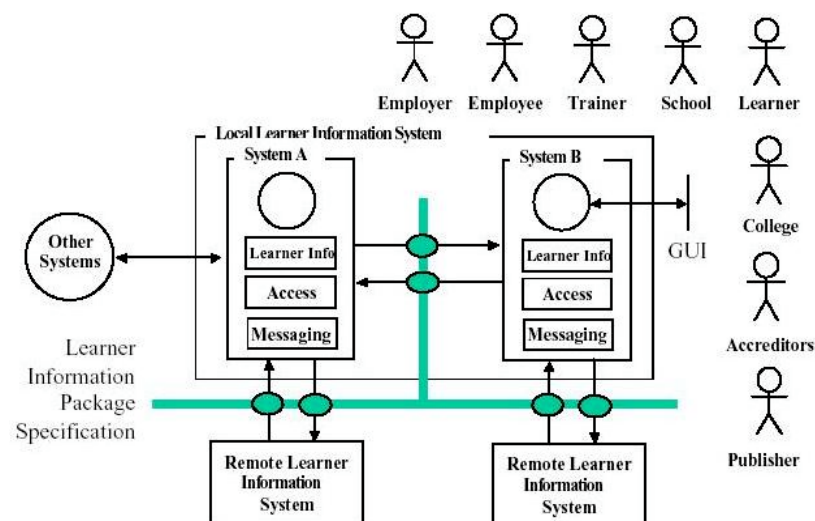
- Almacenamiento y manipulación de la historia, objetivos y logros del aprendizaje.
- Introducir al estudiante en una experiencia de aprendizaje.
- Descubrir a los estudiantes oportunidades de aprendizaje.

La especificación soporta el intercambio de la información del estudiante entre sistemas de gestión de aprendizaje, sistemas de recursos humanos, sistemas de información de estudiantes, sistemas de empresas e-learning, sistemas de gestión de conocimiento, depósitos del currículum vitae y otros sistemas usados en el proceso de aprendizaje. El LIP no esta enfocado a las peticiones de información o al mecanismo de intercambio.

14.10.2. Sistemas de Información del Estudiante

UN SISTEMA

La especificación LIP introduce la arquitectura base del sistema de información del estudiante. Los componentes subrayados (círculos) y las estructuras de datos (rectángulos finos) y los actores se muestran en la siguiente figura:





Los componentes clave son:

- **Sistema local de información del estudiante:** un servidor (es) local que es accesible directamente desde la correspondiente comunidad de usuarios.
- **Sistema remoto de información del estudiante:** sistema distribuido de información, las diferentes partes del _learner information_ pueden ser almacenadas en distintos servidores.
- **Otros sistemas:** otros sistemas pueden ser interconectados a los servidores de información del estudiante (e-mail). Los interfaces de estos sistemas se encuentran fuera del alcance de esta especificación.
- **Estructuras de datos:**
 - Learner Info: la información actual del estudiante.
 - Access: los derechos de acceso a la información del estudiante (quien puede ver que)
 - Messaging: el protocolo de mensajería usado par implementar el perfil actual de intercambios.
- **Actors:** los diferentes roles de los usuarios que acceden al servidor. Pueden existir más actores de los que aparecen en esta figura. Los actores acceden al sistema mediante una GUI (Interfaz Gráfico de Usuario).

INFORMACIÓN DEL ESTUDIANTE

Es una colección de información sobre un estudiante (individual o un grupo de estudiantes) o un productor de contenidos de aprendizaje (creadores, proveedores o vendedores). La especificación LIP se dirige a la interoperabilidad entre sistemas de información del estudiante basados en Internet con otros sistemas que soporten el entorno de aprendizaje de Internet. Esta especificación intenta definir un conjunto de paquetes que puedan ser usados para importar y exportar datos desde un servidor. Es responsabilidad del sistema permitir al propietario de la información definir que parte de la información puede ser intercambiada con otros sistemas.

El LIP está centrado en información del estudiante, es decir, otra información como actividades administrativas solo importan en el sentido en el que interactúan como actividades educativas. Los tipos habituales de información son:

- **Expediente educativo:** el expediente de rendimiento desde la escuela hasta la universidad. Todos los sistemas educativos conocidos deben ser soportados.
- **Registro de formación:** el registro de las actividades de formación.



- **Registro de la evolución profesional:** registro de las actividades profesionales emprendidas.
- **Currículo Vitae:** un registro personal que incluye referencias a la experiencia en el trabajo, títulos e historial académico. Deben soportarse distintos tipos de CV: académicos, negocio, médicos...
- **Registro del Alumno:** un registro de las actividades de aprendizaje y los éxitos de un individuo, se debe reflejar de manera secuencial (por fechas).
- **Registro del trabajo comunitario:** registro de las actividades comunitarias de un individuo y su correspondiente trabajo y experiencia.

CATEGORÍAS DE INFORMACIÓN DEL ALUMNO

La siguiente tabla hace un repaso de las categorías generales de los datos de la información del estudiante. Cada una de las categorías proporciona ejemplos de los datos que deben ser incluidos.

- **Attributes:** son características sobre los consumidores o productores que afectan al aprendizaje de alguna manera. Algunos atributos son fijos y otros son más variables.
- **Portfolio:** se refiere a las actividades de aprendizaje completadas o en proceso.
- **Learners:** normalmente son estudiantes individuales, pero también pueden ser grupos.
- **Producers:** pueden ser organizaciones o individuos, incluyen 3 categorías generales:
 - **Creators:** Desarrolladores de contenidos de aprendizaje, tales como una sociedad que ha desarrollado una clase o un fotógrafo que proporciona materiales JPEG.
 - **Providers:** La organización que entrega el contenido al estudiante, tales como un departamento de formación, una universidad o un tutor.
 - **Vendors:** una entidad que ofrece productos para ayudar a un entorno de aprendizaje, tales como herramientas software usadas para el desarrollo, sistemas de empresa...



	Learners	Producer
	<ul style="list-style-type: none">• Individual Learners (you and me)• Group Learners	<ul style="list-style-type: none">• Creators• Providers• Vendors
Attributes – Fixed	Identification and location (ID, Name, address, phone, email, web-address) Physical, technical and cognitive characteristics (birth-date, disabilities)	Identification and location (ID, Name, address, phone) Organisation type (public, private, school, business)
Attributes – Variable	Goals, learning plans Temporary conditions Learning preferences	Product lines
Portfolio – Self reported	Work products References Experience and education claims	Sample work Expertise claimed by producer
Portfolio – 3 rd party reported	Transcripts Certifications	Professional qualifications Testimonials

14.10.3. Casos de Uso

Es posible un rango de casos de uso, pero en esta versión, sólo presentamos un número limitado:

- **Perspectiva del Individuo:** el proceso que emprendería un individuo que esta buscando un trabajo y desea incluir su currículo.

Usando los formatos comunes de la especificación IMS LIP, los estudiantes y empleados, pueden enviar sus datos electrónicamente al NVC (National Validation Center). En un futuro las instituciones crearán sus bases de datos compatibles y accesibles por medio de un protocolo compartido en el cual el NVC busque y muestre la información de un individuo dado. Así se podrá comparar fácilmente la información de los distintos aspirantes, usando el mismo formato.

- **Perspectiva del Vendedor:** el proceso que deben soportar los productos que quieren sustituir los sistemas de gestión de recursos humanos y sistemas de administración de estudiantes.
- **UK Higher Education:** información del estudiante que debe ser intercambiada entre organizaciones (por ejemplo, vía Internet).

UK Higher Education es una colección de organizaciones. Hay organizaciones que han crecido durante un largo periodo y otras que han nacido como resultado de las iniciativas políticas del siglo XX. El resultado es un conjunto muy diferente de organizaciones.



Por ejemplo, dos alumnos estudiando un curso con el mismo título en diferentes universidades, pueden estar recibiendo cursos muy diferentes en estructura, contenido y planteamiento. Hasta la fecha muchas universidades han confiado en diversos sistemas de información heterogénea para el intercambio interno de información.

- **Gestión de la carrera:** un registro del rendimiento en la carrera, planes asociados y la supervisión de las metas y objetivos.

Una iniciativa para proporcionar asistencia a todos los ciudadanos en la creación y gestión de carreras y facilitar una vida mas larga de estudiante, el USDoL esta desarrollando un *Career Management Account System* 'CMA' (sistema de información de gestión de carreras). Este prototipo proporcionará un almacén centralizado donde se almacenará toda la información relevante de la carrera.

- **Gestión del grupo de formación:** gestión de los grupos sometidos a un entrenamiento, por ejemplo, un grupo de empleados que está recibiendo una formación de introducción.

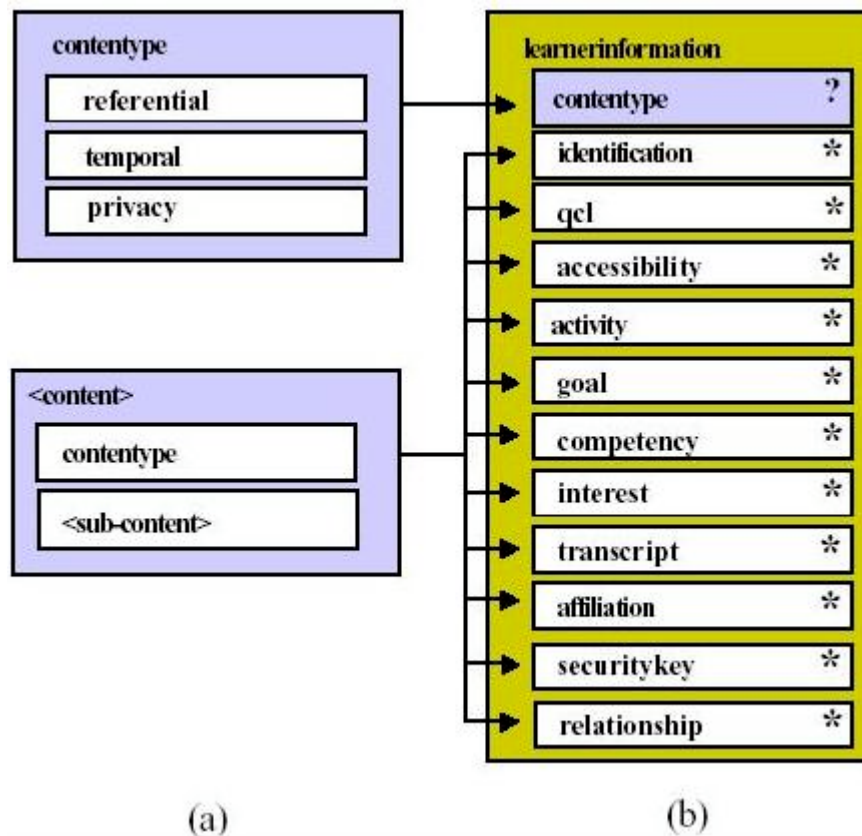
14.10.4. Modelo de Información Básico

El modelo completo de información tiene 3 componentes:

- **Paquete de información del estudiante:** estructuras que contienen la información de los estudiantes.
- **Intercambio de información del estudiante:** ayuda para agregar paquetes, transacciones y protocolos de mensajes.
- **Consulta de la información del estudiante:** es el mecanismo de consulta por el cual se pueden recuperar los componentes de la información del estudiante de acuerdo con el criterio de búsqueda.

PAQUETE DE INFORMACIÓN DE USUARIO

La siguiente tabla muestra la relación entre los elementos *learnerinformation*. Estas estructuras pueden tener una subestructura recursiva. La subestructura *_atomica_* es el nivel mas bajo para el que hay un único identificador de la referencia, es decir, el nivel más bajo para el que el *contentype* existe. Cada una de las 11 estructuras pueden aparecer varias veces en la estructura de información del estudiante, por ejemplo: deben existir distintas entradas para cada título, certificado y licencia.



En la figura (b) se pueden ver los tipos de información considerados fundamentales para las estructuras de datos de información del usuario y la información de contenido usada para almacenar la información describiendo el contenido. La información de contenido, **contentype**, que se muestra en la figura (a) consiste en lo siguiente:

- **Referential**: información que permite a los componentes de datos identificarse de manera única. Esta referencia se usa para asegurarse de que futuras acciones sobre los datos sean relacionadas con la creación original de los datos.
- **Temporal**: información que se usa para describir por ejemplo la creación de la información (date/time stamp)
- **Privacy**: información que se usa para asegurar la integridad de los datos y describir su privacidad.

Es importante...

- La información del estudiante puede consistir en una sola subestructura atómica, es decir, varias subestructuras de una de las 11 estructuras básicas de datos. Esto significa que solo necesitan ser transferidos los datos relevantes.



- La información del estudiante puede describir un individuo o una organización. Se debe adoptar la especificación IMS Enterprise para intercambio de información basada en grupos.

INTERCAMBIO DE INFORMACIÓN

El intercambio de instancias LIP requiere...

- **Empaquetamiento de múltiples instancias LIP:** esto es interesante si se van a intercambiar múltiples instancias LIP entre sistemas. Se recomienda adoptar la especificación IMS Content Packaging para esta agrupación.
- **Gestión de Transacciones:** este es el mecanismo por el que las instrucciones relacionadas con los datos incluidos dentro del LIP pueden ser procesadas por el sistema receptor. Los sistemas de comunicación deben agregar un conjunto de transacciones, por ejemplo: crear un registro, borrar un registro, actualizar un registro...
- **Mensajería:** El mecanismo apropiado de mensajería, que debe incluir los protocolos para todas las transmisiones actuales de datos.

CONSULTA DE LA INFORMACIÓN DEL ESTUDIANTE

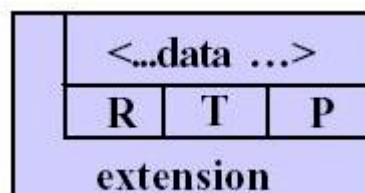
Para poder realizar consultas se requiere...

- La definición de un lenguaje de consultas adecuado.
- El empaquetamiento de los resultados producidos por la consulta. Se recomienda adoptar la especificación IMS Content Packaging.

14.10.5. Empaquetamiento de la información del estudiante

ESTRUCTURAS DE DATOS PRINCIPALES

Las estructuras de datos principales se definen usando la siguiente representación



- **gt...data...lt** La información del estudiante actual. Pueden ser cualquiera de las siguientes estructuras: interest, qcl, goal, transcript...
- **R** --- Referential: La estructura de información puede usarse para almacenar datos que identifiquen el _data_ de manera única.



- **T** --- Temporal: la estructura de información puede usarse para almacenar datos basados en el tiempo sobre el `_data_`, por ejemplo: la fecha de creación del `_data_`
- **P** --- Privacy: La estructura de información puede usarse para almacenar datos privados y para asegurar la integridad de los datos.
- **Extensión:** se puede usar para mantener las características específicas de implementación.

Las 11 estructuras principales de datos son:

- **Identification:** contiene todos los datos de un individuo u organización específico. Incluye datos como: nombre, dirección...
- **Accessibility:** describe las capacidades del estudiante para interactuar con el ambiente de aprendizaje.
- **Qcl:** son los títulos, certificados y licencias otorgados al estudiante, es decir, los productos reconocidos formalmente de su historia académica y laboral. Se puede usar una estructura `_qcl_` diferente para cada título...
- **Activity:** información académica, laboral y de servicios (militar, voluntarios...), excluyendo los oficiales. Se puede incluir información sobre cursos realizados... Podemos usar una estructura diferente para cada entrada.
- **Goal:** es una descripción de los objetivos personales y aspiraciones. Un objetivo se puede definir en términos de sub-objetivos. Se puede usar una estructura diferente para cada entrada.
- **Competency:** describe las habilidades que el estudiante ha adquirido. Se puede usar una estructura `_competency_` diferente a través de un mecanismo de referencia externa. Interest: describe hobbies y otras actividades de recreo. Se pueden describir en distintas estructuras.
- **Interest:** describe hobbies y otras actividades de recreo. Se pueden describir en distintas estructuras
- **Transcript:** se usa para almacenar los expedientes académicos de una institución, puede contener distintos niveles de detalle.
- **Affiliation:** almacena las afiliaciones del estudiante a las distintas organizaciones, pueden incluir grupos educativos (clases). Para el intercambio se recomienda usar técnica de la especificación IMS Enterprise.
- **Securitykey:** almacena los passwords y los códigos de seguridad que se deben usar a la hora de comunicarse con el estudiante. Se puede usar una estructura diferente para cada clave.

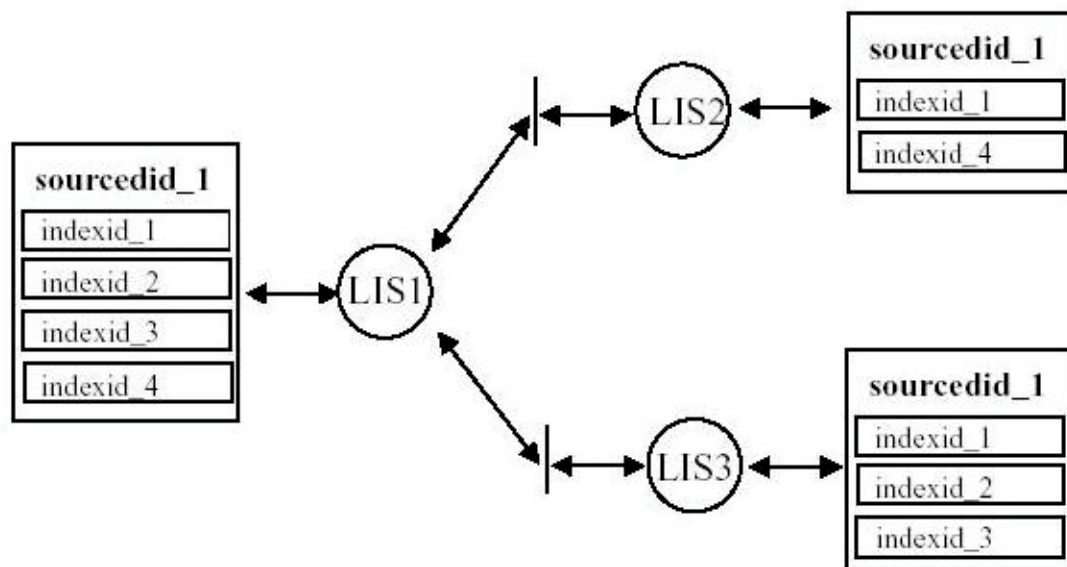


- **Relationship:** describe las relaciones con otras estructuras de datos. Toda la información de relaciones debe ser borrada de las otras estructuras para almacenarlo en un único lugar. También se puede usar para describir las relaciones de mapeo que deben usar los sistemas de comunicación.

SISTEMAS DISTRIBUIDOS

El LIP soporta intercambio de información entre sistemas de información distribuidos. Esto se consigue con un sistema de referencia flexible que se pueda usar para identificar los registros y las estructuras de datos del estudiante en un registro. Los dos mecanismos son:

- **Sourcedid:** el identificador de registro de información del estudiante. Consiste en una etiqueta fuente, única para la fuente responsable de crear la información, y su identificador del registro dentro de esa fuente. La fuente es responsable de que cada registro de información tenga un identificador único.
- **Indexid:** se puede asociar un índice, único dentro del registro de información, a las 11 estructuras de datos básicas y las subestructuras asociadas usadas para almacenar la información. Esto permite en futuras operaciones identificar la pieza apropiada y así transferir solo esta pieza en lugar de toda la información. El indexid es un puntero y sólo el sistema que lo use puede mantener la tabla de mapeo entre el indexid y la dirección local en la base de datos.



En este sistema vemos 3 servidores de información. La información consiste en 4 estructuras (de indexid_1 a indexid_4) y la fuente, LIS1, tiene asignado un sourcedid _sourcedid_1_. LIS1 ha intercambiado su información con los otros dos servidores LIS2 y LIS3.



ESCALABILIDAD

Dentro del sistema de información del estudiante puede haber intercambio como:

- un registro completo de información.
- un registro parcial _ esto es lo más probable.

En ambos casos la información intercambiada puede ser para un estudiante solo o para cientos de usuarios. La información puede incluir productos de educación, entrenamiento o historia del trabajo del individuo y estos productos pueden ser gráficos (ilustraciones de alta resolución), vídeo... En tales casos los productos mismos pueden ser muchos megabytes/gigabytes. Cada uno de estos asuntos se debe tratar para asegurarse de que el IMS LIP sea escalable en términos de almacenaje y accesibilidad eficientes para millones de registros. Los mecanismos para la escalabilidad soportados por el LIP son:

1. Intercambio granularidad que requiere solo incluir la información pertinente. Esto se alcanza rompiendo la información del estudiante en paquetes claramente definidos que podrían extenderse desde un nuevo único nombre a un film codificado digitalmente de 90 minutos. Cada paquete se identifica unívocamente dentro de una estructura de información únicamente identificada.
2. La estructura de intercambio consiste en uno o más estructuras de información independientes o relacionadas. Esto significa que la transferencia de información se puede usar para aprovecharse de las estructuras de paquetes más eficientes.
3. Referencias externas a materiales a través del uso de URIs y definiciones de entidades XML así como apoyo para embeber el material directamente dentro del paquete.

PRIVACIDAD Y PROTECCIÓN DE DATOS

La privacidad e integridad de los datos que se intercambian es esencial. Mientras los detalles de la arquitectura de seguridad que se están usando para soportar el sistema de información del estudiante fuera del alcance de esta especificación es importante proporcionar mecanismos que puedan usarse para soportar la implementación de cualquier arquitectura adecuada. El LIP tiene dos mecanismos:

1. soporte para la inclusión de información que puede usarse para describir el nivel de privacidad, derechos de acceso e integridad de los datos. Esto se define como la estructura de privacidad y protección de datos.
2. Soporte para la información del estudiante que será usada para permitir la transferencia de los datos segura y/o autenticada. Esto se describe como las claves de seguridad del estudiante.

Meta-estructura de Privacidad y Protección de Datos.

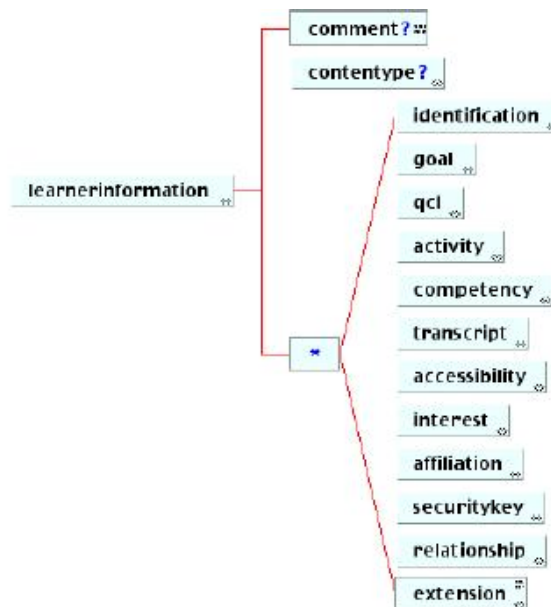
Entro de la estructura de árbol de la información cada nodo y hoja tiene asociado un conjunto de información de privacidad (el uso de estos campos es opcional). La granularidad de la información que puede ser intercambiada se define por el más pequeño conjunto de datos en los cuales no hay más datos de privacidad independientes. La naturaleza de la privacidad de los datos está más allá del alcance de esta especificación.

Claves de Seguridad del Estudiante

Las claves de seguridad del estudiante incluyen sus propias claves públicas para encriptación pública, passwords para acceso a la información (electrónicos y verbales) y firmas digitales usadas para asegurar la autenticidad de los datos. La estructura detallada para estas claves no se definirá pero estos datos serán soportados en la estructura principal 'securitykey'.

14.10.6. Descripción Conceptual de los Objetos de Datos

Los elementos principales de la información del estudiante se muestran en la siguiente figura.





Extensión y Extensibilidad

Un requisito indispensable para la especificación está soportado, cuando es apropiado, por las extensiones. Estas extensiones toman 3 formas:

- Extensiones de definición: extensiones a la especificación de acuerdo con las políticas de los sistemas que se comunican. La estructura interna de los `_resultados_` es un ejemplo de este tipo de extensión. La organización interna de estas estructuras está fuera del alcance de esta especificación pero la manera en la que están estructuradas y usadas está limitada por la especificación.
- Extensiones de vocabulario: extensiones que permiten que los vocabularios básicos sean ampliados. Estos vocabularios están asignados a varios objetos de datos y son usados para definir el tipo de información que contienen. Los vocabularios básicos están definidos dentro de un conjunto de ficheros mantenidos por IMS. Las extensiones de vocabularios pueden estar incluidas en una lista de strings o referenciadas usando una URI.
- Extensiones funcionales: extensiones que se han incluido para asegurar que los usuarios de la especificación puedan añadir funcionalidad que en otro caso estaría excluida de la especificación (en la siguiente descripción tabular está denotado por la estructura 'extension').

Descripción Tabular del Paquete de Información del Estudiante

Las tablas de esta sección proporcionan una descripción conceptual e informativa de los elementos en los objetos de datos. Las columnas en estas tablas se refieren a:

No: el número del elemento. Un elemento puede estar compuesto por sub-elementos. La numeración del esquema refleja estas relaciones

Name: el nombre descriptivo del elemento

Explanation: descripción del elemento.

Required: indica si el elemento es requerido:

- M= elemento obligatorio que debe ser incluido en el objeto de datos, si el elemento al nivel mas alto esta incluido
- C= elemento condicional. Existe o no dependiendo de los valores de otros elementos
- O= elemento opcional

Mult: multiplicidad del elemento

- Blank: instancia simple



- Number: máximo numero de veces que el elemento se repite
- N: sin limite, permitidas múltiples ocurrencias

Type: descripción de las reglas de formación del elemento.

- ID= elemento usado para identificar de manera única un objeto
- Code= valor del elemento que se obtiene de una lista de códigos
- Descripción= elemento descriptivo, lenguaje humano
- Flag= flag binario
- Enumerated= lista de opciones no numéricas predefinidas, por ejemplo: la lista definitiva de objetos.

En todos los campos se puede usar el conjunto de caracteres especificado por ISO 10646. El tipo puede incluir una descripción del conjunto de valores válidos para los sub-elementos.

Note: información descriptiva adicional sobre el elemento.

Tablas...

- [Learner Information Package Data Objects](#) describe los objetos de datos usados en la construcción del paquete de información del alumno
- [Estructura de Datos 'identification'](#)
- [Estructura de Datos 'accessibility'](#)
- [Estructura de Datos 'goal'](#)
- [Estructura de Datos 'qcl'](#)
- [Estructura de Datos 'activity'](#)
- [Estructura de Datos 'competency'](#)
- [Estructura de Datos 'interest'](#)
- [Estructura de Datos 'affiliation'](#)
- [Estructura de Datos 'transcript'](#)
- [Estructura de Datos 'securirykey'](#)
- [Estructura de Datos 'relationship'](#)
- [Estructuras de Datos Comunes](#)
- [Extensiones funcionales, <extension>, soportadas](#)



14.10.7. Vocabularios & Taxonomías IMS LIP

Dentro del LIP hay varios vocabularios especiales que se requieren para definir el tipo específico de la información que se incluye. Estos vocabularios y los nombres de los ficheros IMS asociados se listan en la siguiente tabla:

No	Source Element	File Name	Default Vocabulary
1	activity	imslipvlp0_activity.txt	Work, Service, Education, Training, Military
2	address	imslipvlp0_address.txt	Work, Permanent, Private, Temporary, Mailing, Campus, Billing
3	affiliation	imslipvlp0_affiliation.txt	Professional, Personal ¹ , Military, Civic
4	agent	imslipvlp0_agent.txt	Parent, Guardian, Proxy, Aide, Advisor, Tutor, Mentor, Sponsor
5	agentdomain	imslipvlp0_agentdomain.txt	Legal, Medical, Financial, Accessibility, Educational
6	contactinfo ⁴	imslipvlp0_contactinfo.txt	Private, Work, Campus
7	date	imslipvlp0_date.txt	Effective ⁵ , Birth, Start, Finish, Expiry, Death, Update, Create, Renewal, Delete, Publish, Award, Enrol, Join
8	definition	imslipvlp0_definition.txt	Class, Course, Curriculum, Module, Topic, Unit
9	demographics	imslipvlp0_demographics.txt	Adult, Mature, College, Primary, Secondary, Preschool, Nursery, University, Vocational, Enrichment, Graduate, Professional, Technical
10	disability	imslipvlp0_disability.txt	For further study in V2.0 ⁶
11	eligibility	imslipvlp0_eligibility.txt	For further study in V2.0 ⁷
12	evaluation ⁸	imslipvlp0_evaluation.txt	QTI_Assessment, QTI_Section, QTI_Item
14	goal	imslipvlp0_goal.txt	Work, Education, Personal
15	interest	imslipvlp0_interest.txt	Recreational, Vocational, Domestic
16	language	imslipvlp0_language.txt	Use the ISO Standard terminology
17	name	imslipvlp0_name.txt	Contact, Full, Alias, Maiden, Preferred, Former
18	organization	imslipvlp0_organization.txt	Professional, Employer, Government, Recreational, Educational, Training, Military
19	partname	imslipvlp0_partname.txt	Particle ⁹ , Prefix, Suffix, Given, Middle, Surname, Nickname, Last, First, Family, Maternal, Paternal, Initials
20	preference	imslipvlp0_preference.txt	Cognitive, Physical, InputTech, OutputTech
21	privacy	imslipvlp0_privacy.txt	Creator, Owner, Steward, Learner, Default, [All] ¹⁰
22	product	imslipvlp0_product.txt	Exam, Coursework, Portfolio, Participation
23	qcl	imslipvlp0_qcl.txt	Qualification, Certification, Licence, Degree
24	relationship	imslipvlp0_relationship.txt	Activity, Accessibility, Affiliation, Competency, Goal, Identification, Interest, Qcl, Securitykey, Transcript
25	representation	imslipvlp0_representation.txt	Photo, Voice, Biometric, Signature
26	role	imslipvlp0_role.txt	Administrative, Executive, Officer, Representative, Member
27	securitykeys	imslipvlp0_securitykeys.txt	Password, Certificates, PIN, Username
28	status	imslipvlp0_status.txt	Active, Inactive, Retired, Completed, InProgress, Pending, Expired
29	temporal	imslipvlp0_temporal.txt	Expiry, Creation, Update, Purge
30	testimonial	imslipvlp0_testimonial.txt	Academic, Personal, Work, Military, Service
31	transcript	imslipvlp0_transcript.txt	Academic, Vocational, Training



Los nombres de los ficheros por defecto, se usan cuando el atributo `_sourcetype_` tiene asignado el valor `_imsdefault_`.

14.10.8. Descripciones de los Meta-datos

Hay un tipo de meta-dato incluido explícitamente en esta especificación:

- la estructura `<evalmetadata>` (se localiza dentro de la estructura `<evaluation>`) se usa para almacenar los meta-datos asociados con el mecanismo usado para permitir la evaluación. Los meta-datos usados aquí deben salir del mecanismo de evaluación asociado, por ejemplo usando la especificación del IMS QTI para exámenes, secciones e ítems según lo definido por la especificación QTI.

Inclusión de meta-datos IMS

Los elementos IMS Meta-data no están soportados directamente dentro del IMS LIP. Esto es porque las instancias IMS LIP serán empaquetadas usando la especificación IMS Content Packaging. Esta especificación ya define como deben incluirse los meta-datos.

14.10.9. Conformidad

El propósito de esta declaración de conformidad es proporcionar un mecanismo a los clientes para comparar a los vendedores de contenidos y herramientas de información del estudiante. No es obligatorio para un vendedor apoyar todas las características para presentar conformidad, sin embargo, debe detallar su nivel de conformidad con un 'Conformance Statement'.

Conformance Statement

La declaración de conformidad se presenta en 12 tablas:

- [<learnerinformation>](#)
- [<accessibility>](#)
- [<activity>](#)
- [<affiliation>](#)
- [<competency>](#)
- [<goal>](#)
- [<identification>](#)
- [<interest>](#)



- [<qcl>](#)
- [<relationship>](#)
- [<securitykey>](#)
- [<transcript>](#)

En cada tabla los 'check-boxes' relevantes están marcadas para indicar que soporta la propiedad correspondiente.

[Para descargar los ficheros del estandar originales pincha aquí](#)

14.10.10. Ver evaluación en Internet

14.11. IMS DIGITAL REPOSITORIES INTEROPERABILITY SPECIFICATION

14.11.1. Introducción

Este documento constituye el modelo de información para la fase 1 de esta especificación (DRI). El propósito de esta especificación es dar recomendaciones para la interoperabilidad de las funciones más comunes. Estas recomendaciones deben implementarse a través de los servicios para permitir presentar un interfaz común. Trata de utilizar esquemas ya definidos en otra parte (IMS Meta-data y Content Packaging), en lugar de introducir uno nuevo.

En el más amplio nivel, esta especificación define almacenes digitales como cualquier colección de recursos accesibles por medio de una red sin un conocimiento previo de su estructura. Los almacenes deben mantener valores actuales o los meta-datos que los describan. Los valores y sus meta-datos no necesitan estar en el mismo almacén.

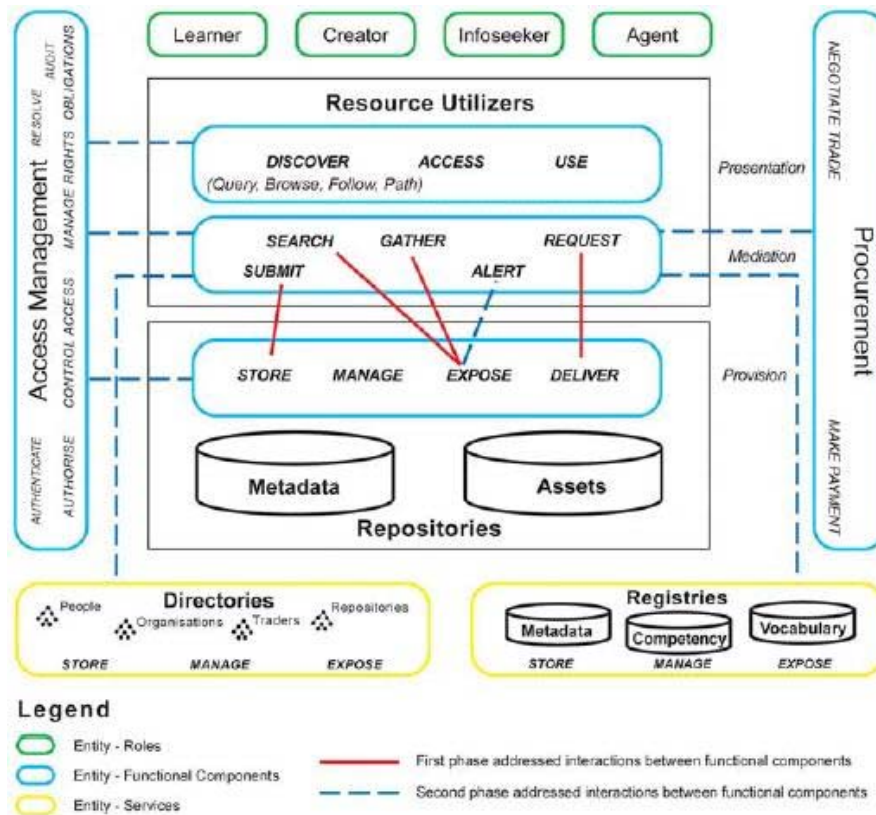
14.11.2. Arquitectura Funcional

El siguiente diagrama representa la arquitectura funcional del DRI. Las interacciones se representan con las líneas rojas (sólidas). El diagrama muestra 3 tipos de entidades que definen el espacio donde interactúan e-learning, almacenes digitales y Servicios de información.

Las 3 entidades son:

- Papeles (Estudiante, Creador, Agente, Buscador de Información)
- Componentes funcionales para usuarios de recursos, almacenes, acceso a la gerencia y servicios de aprovisionamiento.

- Servicios, como registros y directorios (no es parte del alcance de la fase 1 del DRI)

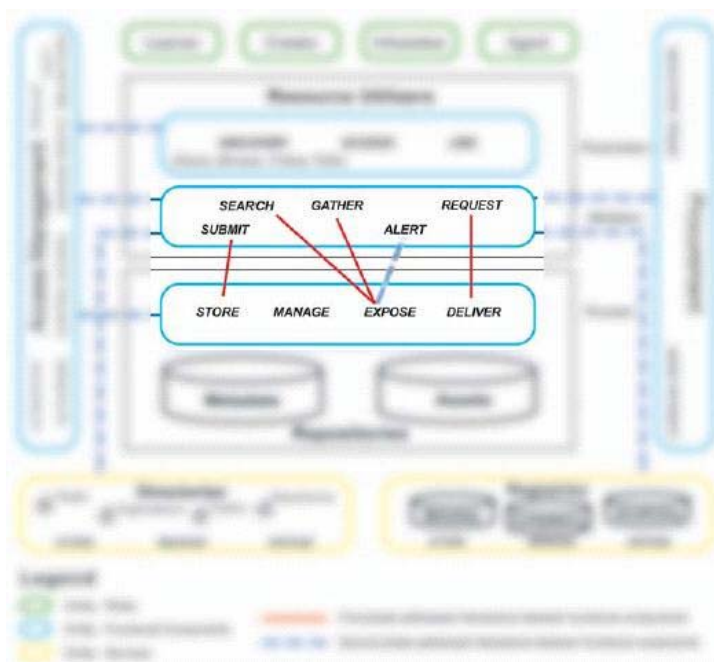


Las líneas rojas, entre un número de funciones entre el recurso Utilizers y los almacenes, indican las interacciones entre los principales componentes funcionales que soportan interoperabilidad, incluyendo:

- **SEARCH, GATHER, (ALERT)/EXPOSE:** BUSCAR, RECOGER, (ALERTAR)/EXPONER
- **REQUEST/DELIVER:** SOLICITAR/ENTREGAR
- **SUBMIT/STORE:** PROPONER/ALMACENAR
- **DELIVER/STORE:** ENTREGAR/ALMACENAR (entre dos almacenes).

Nota: ALERT es una función principal, pero no es parte de esta versión de la especificación.

El grupo del proyecto DRI está centrado en estas funciones principales.

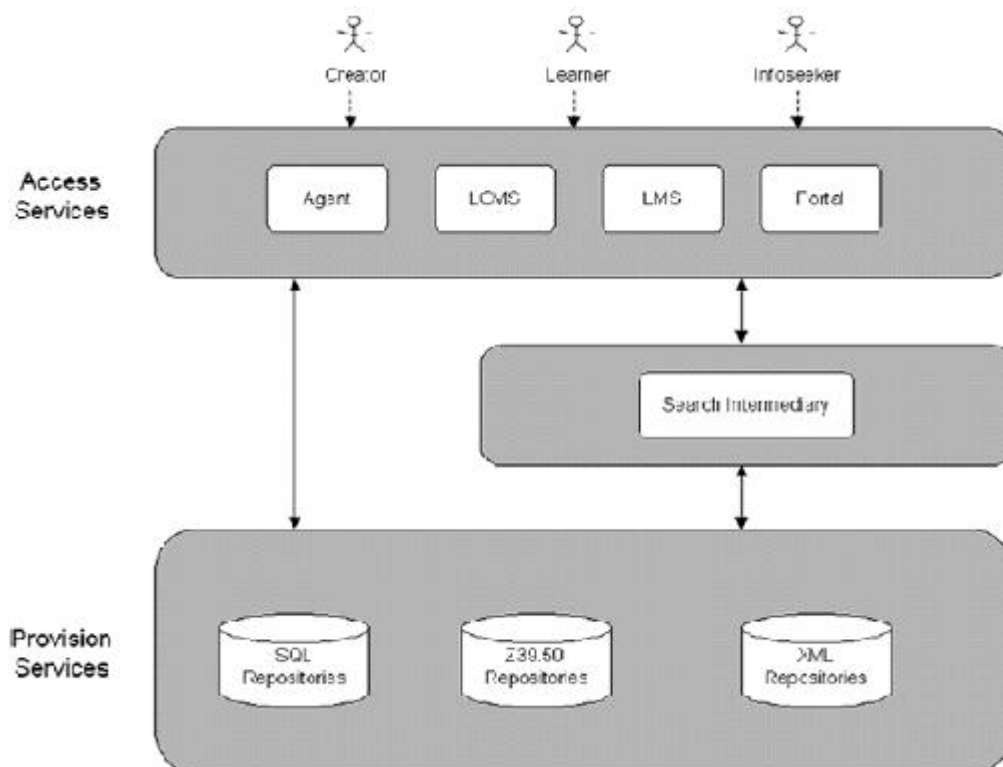


14.11.3. Modelo de Referencia

El siguiente diagrama proporciona una vista simplificada de los sistemas del dominio digital del almacén de los componentes que incorporan las funciones identificadas en el diagrama anterior. Hay dos tipos de almacenes representados en la siguiente figura

- Sistemas que reflejan la práctica establecida para la interoperabilidad de los almacenes.
- Almacenes que son capaces de implementar las recomendaciones XQuery y SOAP, según lo propuesto en esta especificación.

La sección 2 (Arquitectura Funcional) describe 4 roles que juegan los usuarios de los almacenes digitales: Creador, Estudiante, Buscador de Información y Agente. En la siguiente figura se muestran los usuarios jugando 3 de estos roles y las típicas aplicaciones de software en las que interactúan.



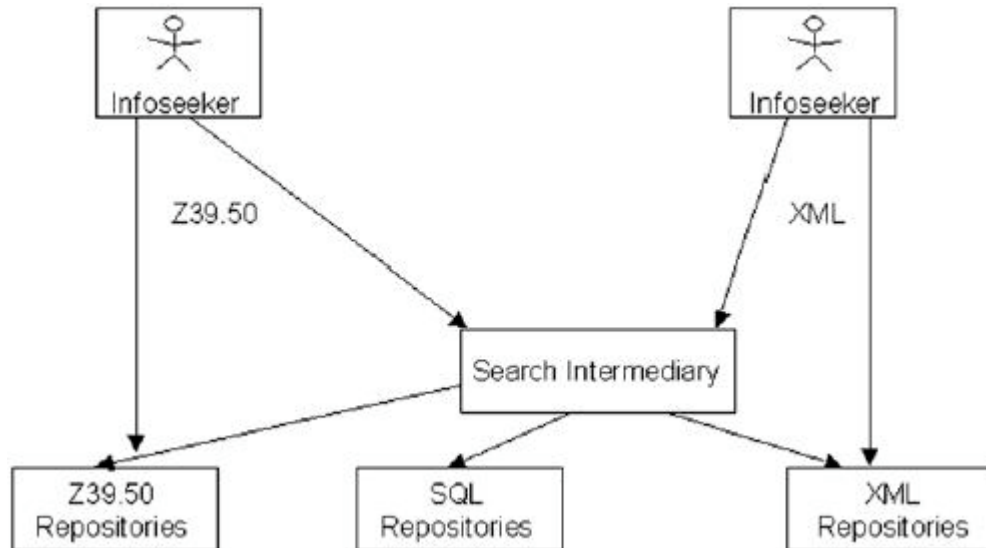
El objetivo de esta especificación es permitir acceso generalizado al contenido del almacén de todos los tipos en el contexto e-learning por aplicaciones como LMS (Learning Management Systems), LCMS (Learning Content Management Systems) y portales de búsqueda, como se muestra en el modelo de referencia (en la figura anterior). Estas aplicaciones de software se usan como ejemplos comunes para la industria e-learning.

Buscar contenidos, cuando hay múltiples almacenes de contenido donde buscar, es un problema complejo. El problema se agrava cuando el almacén tiene representaciones de meta-datos y métodos de acceso heterogéneos. El modelo de referencia introduce un componente opcional intermedio que pueda satisfacer una de las 3 funciones que simplifican este problema:

- Una función de traducción es capaz de traducir un formato de búsqueda en otro y es entendida por el IMS y los almacenes existentes.
- Una función de agregación que recolecta IMS y/o otros meta-datos de múltiples almacenes y que convierte estos meta-datos en disponibles para la búsqueda.
- Una función Federator que pasa una consulta de búsqueda a los múltiples almacenes y manejan sus respuestas.

14.11.3.1. Search/exponse (buscar/exponer)

El modelo de referencia Search define la búsqueda de meta-datos asociados con el contenido expuesto por los almacenes. El modelo de referencia se muestra en la siguiente figura:



Hay 2 características dominantes en este modelo. Primero, soporta un amplio rango de configuraciones para dirigir las búsquedas. Eso puede ofrecer amplias experiencias basadas en tecnologías y orientadas a la comunidad para buscar el contenido digital universal. Segundo, ofrece una capa opcional de mediación para permitir la consulta de fuentes de meta-datos heterogéneas y distribuidas.

Recomendaciones para el lenguaje de consulta

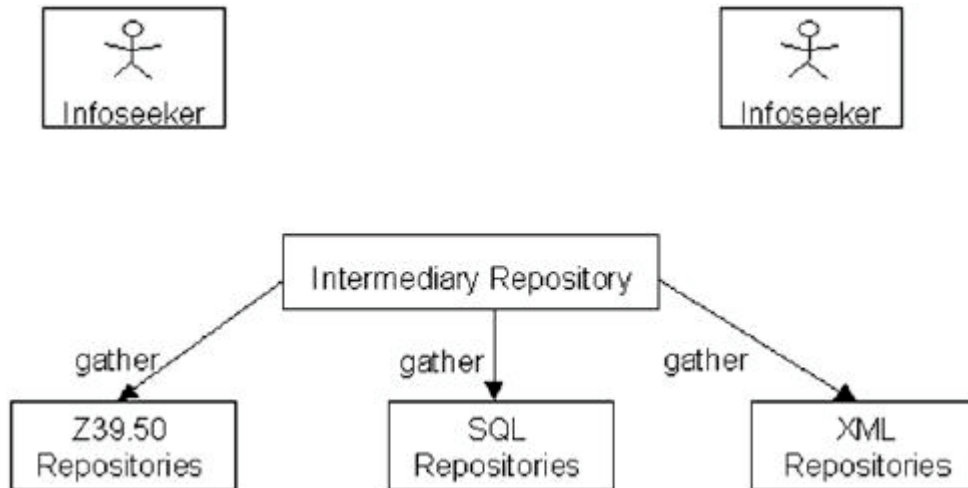
1. XQuery para buscar meta-datos IMS (XML). XQuery ha sido desarrollado por W3C. Tiene una gramática bien desarrollada, y están apareciendo varias implementaciones comerciales.
2. Z39.50 para buscar información de biblioteca. Esta búsqueda proporciona una gramática para buscar almacenes Z39.50 directamente o a través de un motor intermedio de búsqueda.

Hay 3 tipos de mensajes SOAP con o sin adjuntos:

- RPC
- Messaging
- Session

14.11.3.2. Gather/exponse (recoger/exponer)

El modelo de referencia Gather define la solicitud de meta-datos expuestos por almacenes y la agrupación de los meta-datos para usarlos en búsquedas posteriores y para crear un nuevo almacén de meta-datos. El almacén se convierte en otro disponible para funciones Search/Exponse, Alert/Exponse. El modelo de referencia se muestra en la siguiente figura.



El componente Gather puede interactuar con los almacenes en 1 de los dos caminos posibles. Solicita activamente meta-datos (creado nuevamente, actualizado o borrado) de un almacén, o contribuye con un servicio de notificación de meta-datos (creado nuevamente, actualizado o borrado) proporcionado por el almacén (pull), o por un adaptador externo a éste que permita mensajería entre el almacén y otros usuarios (push).

Recomendaciones para "Pull" Gather

La iniciativa de archivo abierto (Open Archive Initiative: OAI) proporciona un modelo simple para "Pull" Gather. Los meta-datos OAI realizan una búsqueda periódica de almacenes objetivo y recuperan los meta-datos de un rango de fechas. La fecha es el principal criterio usado en este modelo, y requiere un elemento dentro de los meta-datos que contenga la fecha en la que fue añadido o actualizado. Esto tiene la ventaja de ser muy simple y puede ser efectivo recolectando meta-datos.

Para usar el modelo OAI en el contexto IMS, un elemento dentro de la especificación IMS Meta-Data debe estar definido para proporcionar la información de la fecha que se requiere para permitir al sistema Gather determinar que meta-datos son añadidos o actualizados desde la última vez que fueron recuperados de este almacén. Hay que darse cuenta de que el sistema Gather es responsable de mantener la información relativa a la fecha. Hay ediciones trabajando con el modelo OAI, con estructuras de meta-datos que no contienen el elemento fecha requerido.



Actualmente OAI usa mensajes XML sobre HTTP. La documentación del protocolo 1.1 del OAI esta disponible en: <http://www.openarchives.org/OAI/openarchivesprotocol.html>

Otra opción para el Pull Gather es recoger periódicamente todos los registros de meta-datos de todos los almacenes objetivo. Es una utilización ineficiente de los recursos ya que millones de registros serían puestos en la red repetidas veces.

Recomendaciones para "Push" Gather

Es un caso especial de alerta. Cuando se añade o actualiza un nuevo registro de meta-datos, los almacenes pueden enviar un mensaje de aviso. Esto puede ser un mensaje simple diciendo que un nuevo meta-dato existe en este almacén (o puede ser el meta-dato completo), el cual puede ser después incorporado en el almacén de meta-datos y estar disponible para la búsqueda.

Este mecanismo también puede estar provisto de un adaptador externo a un almacén particular. Este adaptador puede enviar meta-datos simultáneamente a los "mezcladores" intermedios cuando un nuevo contenido sea añadido. También puede gestionar la traducción de mensajes entre la red y las capacidades nativas del almacén.

14.11.3.3. Alert/expose (alertar/exponer)

El grupo de proyecto DRI contempla la función Alert como un posible componente de un almacén digital o un servicio intermedio para agregar y se prevé que el e-mail/SMTP pueda proveer esta funcionalidad. Sin embargo, la función Alert se contempla fuera del alcance de la Fase 1 de la especificación DRI.

14.11.3.4. Submit/store (proponer/almacenar)

Se refiere a la manera en la que un objeto es llevado al almacén desde una localización (accesible por red) dada, y como el objeto será representado en ese almacén para su acceso. La localización desde donde se mueve un objeto puede ser otro almacén, un sistema de gestión de aprendizaje, una unidad de un desarrollador o cualquier otra localización conectada a la red.

En el caso de los almacenes desarrollados más recientemente que tratan específicamente con los objetos de aprendizaje, esta especificación hace referencia a la IMS Content Packaging Specification. Esta especificación define 'interoperabilidad entre sistemas que desean importar, exportar, agrupar y desagrupar paquetes de contenido'. Un Content Package consiste en un paquete comprimido en un fichero (preferiblemente un fichero .zip), que contiene el objeto de aprendizaje, su registro metadata y el manifest describiendo los contenidos del paquete.

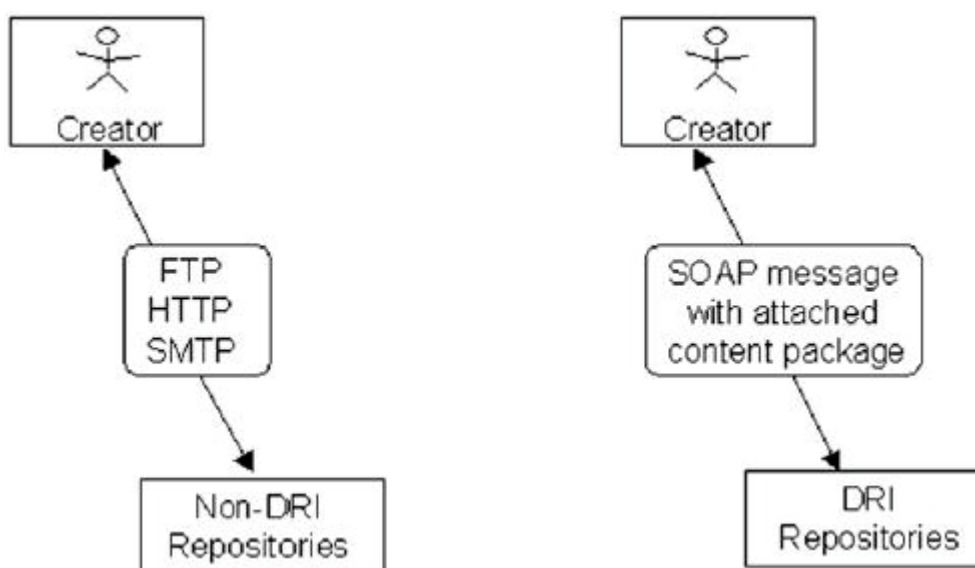
SUBMIT

En el caso del objeto de aprendizaje que sigue la especificación DRI, la función Submit puede satisfacerse a través de la transmisión de un IMS Content Package usando mensajes SOAP con archivos adjuntos.

STORE

La función Store se entiende simplemente como la capacidad del almacén de preservar un IMS Content Package a cierto nivel de su operación.

La siguiente figura muestra como la funcionalidad Submit/Store alojaría ambos almacenes digitales que utilizan o no las recomendaciones IMS DRI.



14.11.3.5. Request/deliver (solicitar/entregar)

El componente funcional Request permite un recurso que ha localizado el registro metadata por medio de la función Search (y posiblemente vía Alert) para solicitar el acceso al objeto de aprendizaje u otro recurso descrito por el meta-dato. La entrega se refiere a la respuesta del almacén que proporciona acceso al recurso.

Exclusiones del alcance

En un ambiente híbrido, los recursos descubiertos mediante una búsqueda a través del dominio, incluyen potencialmente los recursos que no son objetos de aprendizaje y/o no están disponibles online. La versión 1.0 de la especificación DRI trata solamente del caso de solicitar y entregar online recursos de almacenes de objetos.

Los derechos digitales de gestión, incluyendo la verificación de que el usuario del recurso esta autorizado a acceder al este y la definición de un identificador de objeto para localizar la copia más adecuada no esta contemplados aquí.



Estas funciones se realizan, dentro del modelo funcional, por el servicio de gerencia del acceso.

La verificación de la entrega de los materiales no se incluye en esta versión de la especificación. El proceso del e-comercio y del pago se maneja en otro componente funcional.

El mecanismo Solicitar/Entregar

El punto inicial de la función Request/Deliver es un puntero a la dirección del recurso. Si el registro de meta-datos ha sido localizado por medio de la función Search en los meta-datos IMS, entonces ese puntero estará incluido en el elemento 4.3 <location> del registro metadata. Si el usuario del recurso no está autorizado a acceder a este, entonces los servicios de gestión de acceso deben prevenir el acceso al recurso.

El elemento metadata puede consistir en una lista de direcciones o un método que devuelva una dirección o conjunto de direcciones. Éste puede ser un DOI o una OpenURL. El componente funcional que resuelve, es el responsable de devolver la lista de las direcciones. La dirección devuelta debe resolverse a una URL. Accediendo a esa URL debe inicializarse la solicitud. El protocolo usado para entregar el objeto de aprendizaje variará dependiendo del formato del objeto, pero incluirá:

http: para los materiales online incluyendo html, java y pdf

http: para hacer correr el acceso a los materiales (audio, video, etc.)

ftp: para el acceso a documentos, ejecutables, etc.

14.11.4. Modelo General de Mensajería

El modelo de mensajería se usa para proporcionar comunicación general entre componentes definidos por el modelo de información DRI. El modelo básico de la mensajería consiste en un solo mensaje de la fuente al destino seguido de un mensaje simple de respuesta del destinatario a la fuente. En el futuro se soportan otros modelos de mensajería adicionales.

El mensaje básico tiene 2 partes: la cabeza y el cuerpo del mensaje.

```
message
  header
  body
```



Cabeza del mensaje

Contiene información de direccionamiento y de seguridad (mínima).

La autenticación del mensaje es proporcionada por un elemento 'security'.

```
header
  message type
  destination address
  message authentication
```

Cuerpo del mensaje

Consiste en 0 o mas cargas, con 0 o mas elementos 'audit'

```
body
  payload(s)
  audit element(s)
```

Los elementos 'audit' permiten la adición de información relevante de auditoría a cualquier mensaje. Esto podría incluir información referente al pago, uso, funcionamiento... Estos elementos se acumulan mientras un mensaje se mueve por el sistema. En particular, cuando se devuelve una respuesta, incluye todos los elementos 'audit' de la petición correspondiente.

Seguridad del mensaje

Excepto la autenticación del nivel del mensaje identificada en la cabecera, no se identifican mecanismos explícitos de seguridad en este modelo.

14.11.5. Visión general de alto nivel de los casos de uso de un almacén de aprendizaje

Los siguientes casos de uso DRI describen escenarios donde actores asumen papeles específicos en el almacén de aprendizaje. Antes de explorar los casos de uso, lee detenidamente la sección de actores para entender mejor los papeles que asumen.

ACTORES

Cada actor en el caso de uso, descritos debajo, estará actuando en uno de los siguientes papeles descritos anteriormente: creador, estudiante, buscador de información o agente de software.



1. Papeles de Creador

- Casos de uso - almacén simple
 - **Actor 1:** el creador del curso de formación que esta usando un almacén simple para desarrollar cursos o materiales de referencia relativos a un producto particular para usarlos en materiales de formación en casa o materiales de formación para los clientes.
- Casos de uso - múltiples almacenes
 - **Actor 2:** Alguien trabajando en el contexto de una compañía de edición donde existen varias armas de edición y la persona está intentando obtener materiales de diferentes almacenes para desarrollar nuevos materiales de aprendizaje para vender.

2. Papeles del Estudiante

- Casos de uso - almacén simple
 - **Actor 3:** Empleado o cliente que está usando un almacén simple para encontrar un curso de formación o materiales de referencia que se necesitan para incrementar la competencia individual con respecto a un producto particular.
 - **Actor 4:** Coordinador de formación que asocia competencias con los cursos u objetos de aprendizaje en un almacén.
- Casos de uso - múltiples almacenes
 - **Actor 5:** Alguien que ha comprado materiales de formación creados por el Actor 2 y desea recibir actualizaciones periódicas del material siempre que la compañía lo cambie.

3. Papeles del Buscador de Información

- Casos de uso - almacén simple
 - **Actor 6:** Empleado que está buscando, por medio de un portal, información relativa a un tema particular.
- Casos de uso - múltiples almacenes
 - **Actor 7:** Alguien que está buscando información relativa a un tema que puede estar en varios almacenes.



4. Agentes de Software

- **Actor 8:** Aplicación de software LMS
- **Actor 9:** Aplicación de software LCMS
- **Actor 10:** Portal
- **Actor 11:** Almacén

14.11.6. Casos de Uso

[Crear y Modificar Recursos](#)

[Encontrar Recursos](#)

[Notificación o Modificación de Recursos](#)

[Para descargar los ficheros del estandar originales pincha aquí](#)

14.11.7. Ver evaluación en Internet

14.12. IMS REUSABLE DEFINITION OF COMPETENCY OR EDUCATIONAL OBJECTIVE SPECIFICATION

14.12.1. Introducción

Esta especificación define un modelo de información para describir, referenciar e intercambiar definiciones de competencias, principalmente en el contexto del aprendizaje online y distribuido. En esta especificación la palabra competencia se usa en un sentido muy general que incluye técnicas, conocimientos, tareas y resultados del aprendizaje. Esta especificación muestra una forma de representar las principales características de una competencia, independiente de su uso en un contexto particular. La información principal en esta especificación (RDCEO) es una definición textual no estructurada de la competencia que puede ser referenciada por medio de un identificador único. Esta información puede ser redefinida usando un modelo, definido por el usuario, de la estructura de una competencia.

La especificación RDCEO proporciona un modo de crear interpretaciones comunes de competencias que aparecen como parte de un proyecto de aprendizaje o carrera, como prerrequisitos o resultados del aprendizaje. RDCEO nos da referencias únicas a las descripciones de las competencias u objetivos para incluirlos en otros modelos de información.



La RDCEO está pensada para intercambio entre maquinas, pero la información que contiene está destinada a su interpretación humana. El objetivo de esta especificación no es la agrupación de pequeñas competencias en otras más grandes, no como estas deben ser evaluadas, certificadas, almacenadas o usadas como parte de un proceso. Tampoco especifica como los registros de las competencias asociadas con un individuo son estructuradas, almacenadas o intercambiadas.

14.12.2. Discusión del Modelo de Información

Las competencias están definidas y estructuradas de muchas formas en diferentes comunidades de entrenamiento (ACRL, CASAS, CPA, Mager, NOICC, O*Net, PASS, SCANS, TATS). Esta especificación permite a las comunidades de entrenamiento intercambiar información de acuerdo con el modelo que usan.

La EXTENSIBILITY se puede llevar a cabo definiendo la estructura de la Definición de la Competencia u Objetivo Educativo o incluyendo elementos LOM (Learning Object Metadata) en la parte metadata.

El modelo de información consiste en los siguientes elementos:

- **Identifier:** etiqueta única y global para la competencia u objetivo. Este identificador usa los mismos elementos que el identificador definido en el estándar LOM. Está formado por dos sub-elementos: Catalogue y Entry. Este identificador es suficiente para hacer referencia a la competencia en cualquier otro sistema.
- **Title:** etiqueta de texto obligatoria para la competencia u objetivo. Es el nombre corto de la competencia (legible). Se puede repetir en múltiples lenguajes.
- **Description:** Una descripción de la competencia. Es opcional, no está estructurado y está destinado a ser interpretado sólo por humanos. También se puede repetir en múltiples lenguajes.
- **Definition:** descripción opcional estructurada que permite una definición más completa de la competencia o del objetivo educativo, usando normalmente atributos tomados de un modelo específico de cómo debe ser estructurada o definida (la competencia). El elemento definición proporciona una estructura para incluir una colección de exposiciones STATEMENTS que determine una competencia u objetivo educativo. El autor de la RDCEO es libre de usar el elemento Definition de la forma que mejor describa la competencia u objetivo. Contiene los siguientes sub-elementos:



- **Model Source:** un identificador del modelo o estructura sobre el que está basada la definición. El modelo define qué exposiciones se usan para definir la competencia; define el namespace y el conjunto de valores del contenido para "Statement Name". **Nota:** el valor del Model Source puede especificarse para evitar conflictos con otros nombres fuente; por lo tanto se recomienda que sea una URI. Si es una URI puede apuntar a un documento actual que defina la fuente formalmente, pero no es obligatorio.
- **Statement:** es una descripción de una característica individual de una definición. Una definición debe contener al menos una y generalmente contiene múltiples statements. Contiene las siguientes partes:
 - Statement Id: es un identificador local para el statement dentro del modelo. Es un string y es opcional.
 - Statement Name: es un token que se usa para etiquetar el statement. Se toma de un vocabulario fuente definido en el modelo identificado por el elemento Source.
 - Statement Text: descripción textual, no estructurada y opcional de los aspectos de la competencia referidos por el Statement Name.
 - Statement Token: vocabulario opcional de tokens, junto con un identificador de la fuente del vocabulario.
- **Metadata:** registro metadata opcional que describe la RDCEO. Si se incluye un registro metadata, se recomienda que siga el estándar LOM. **Nota:** El elemento Relation puede usarse para identificar una definición con una versión anterior de la definición, y uno o más elementos Classification pueden usarse para indicar donde encaja en una taxonomía de competencias u objetivos educativos.

Se permite mas de un registro metadata, pero si hay mas de uno cada registro debe ajustarse a una especificación distinta de meta-datos. Una implementación debe aceptar cualquier registro de meta-datos que no pueda interpretar, pero no se requiere interpretar tales registros de meta-datos.

TIPOS DE DATOS, MULTIPLICIDADES Y OBLIGACIONES

Los únicos tipos de datos usados en esta especificación, otros distintos a los usados en el elemento <metadata>, son strings LangStrings y vocabularios fuente (sourced). Todos los strings que representan información interpretable por un humano son del tipo de datos LangString, que incluye un atributo lenguaje. Todos estos strings pueden representarse en múltiples lenguajes.



Los vocabularios fuente son tokens representados como strings. Los tokens se representan sólo en un lenguaje. La codificación permite caracteres de 32-bit. El identificador, título y definición son obligatorios. El resto de los elementos superiores son opcionales. Cada elemento Statement debe contener al menos un elemento StatementText.

El elemento <metadata> puede repetirse. El resto de los elementos superiores deben aparecer al menos una vez o en su mayoría una por lenguaje.

ORDEN DE LOS ELEMENTOS DE DATOS

Todos los elementos en este modelo de datos están intrínsecamente desordenados. El orden de los elementos en el resumen del modelo de datos y el orden de los valores en una lista de valores no tiene ningún significado. Por ejemplo, si el modelo incluye 3 statements, el orden no es significativo. Pueden aparecer en cualquier orden sin perder información.

TAXONOMIAS DE COMPETENCIAS O DEFINICIONES OBJETIVAS

Hay varias formas de clasificar competencias u objetivos educativos. Esta especificación pretende referenciar y catalogar una competencia u objetivo, no de clasificarlo. Sin embargo, se puede querer incluir información de relación y clasificación, lo cual se puede hacer por medio de meta-datos opcionales como se especifica en el modelo de información.



14.12.3. Elementos RDCEO

No	Name	Explanation	Reqd	Mult	Value Space	Datatype	Notes
1	Identifier	A globally unique label that identifies this Definition of Competency or Objective	M	Single			
1.1	Catalog	The name or designator of the identification or cataloguing scheme for this entry. A cataloguing scheme.	M	Single	Repertoire of ISO/IEC 10646-1:2000, as allowed by RFC 2396	Character String (smallest permitted maximum: 1000 characters)	Examples: "ISBN", "ARIADNE", "URI", "http://acme.org/compmodcat"
1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this Definition of Competency or Educational Objective. A namespace specific string	M	Single	Repertoire of ISO/IEC 10646-1:2000, as allowed by RFC 2396	Character String (smallest permitted maximum: 1000 characters)	Examples: "2-7342-0318", "LEAO875", "http://imsglobal.org/dco/1234"
2	Title	Text label of this RCEOD	M	Single *		LangString(smallest permitted maximum: 1000 characters)	Examples: "English proficiency", "Schmiblick failure diagnostic level 4", "Demonstrates conflict resolution skills"
3	Description	Description of the Competency or Educational Objective	O	Single *		LangString(smallest permitted maximum: 2000 characters)	Examples: "Proficiency in written and spoken English and use of English for meaningful oral or written expression.", "Performance of level 4 diagnostic as specified in IETM #SCMBLK007"
4	Definition	A list of statements according to a particular definition model	O	Multiple			
4.1	Model Source	Source of the Model being used	O	Single	Repertoire of ISO/IEC 10646-1:2000, as allowed by RFC 2396	Character String (smallest permitted maximum: 1000 characters)	Examples: "3-part-learning-objective", "http://foo.edu/ref/los.xml"
4.2	Statement		O	Multiple			
4.2.1	Statement ID	A local identifying label for the Statement	O	Single		Character String (smallest permitted maximum: 4096 characters)	
4.2.2	Statement Name	Name of the Statement	O	Single	String		Examples: "Condition", "Action", "Standard", "Outcome", "Criteria"
4.2.3	Statement Text	Text of the statement	O	Single *	LangString		Example: "Given a set of integer numbers in the range 1 to 49"
4.2.4	Statement Token	Token value for the statement	O	Single	Vocabulary defined in definition model	Vocabulary (state)	
5	Metadata	Embedded Metadata about this RDCEO	O	Single			
5.1	RDCEO Schema	Describes the schema that defines and controls this RDCEO	O	Single	Repertoire of ISO/IEC 10646-1:2000, as allowed by RFC 2396	Character String (smallest permitted maximum: 100 characters)	If no schema element is present in a RDCEO instance, its value is assumed to be "IMS RDCEO"
5.2	RDCEO Schema Version	Describes the version of the above schema.	O	Single	Repertoire of ISO/IEC 10646-1:2000, as allowed by RFC 2396	Character String (smallest permitted maximum: 20 characters)	If no schema version element is present in a RDCEO instance, its value is assumed to be "1.0"
5.3	{Additional Metadata}	Additional embedded Metadata describing this RDCEO	O	Multiple	The information contained in this section is defined by the IMS Metadata specification.	Smallest permitted maximum 10	See Best Practice document guidance for Metadata records



* Los elementos del tipo LangString y multiplicidad single deben aparecer al menos una vez por lenguaje, pero pueden aparecer múltiples veces con diferentes atributos de lenguaje.

VALORES MÁXIMOS PERMITIDOS MÁS PEQUEÑOS

En el modelo de información, los valores máximos permitidos más pequeños se definen para:

- *Ítems con múltiples valores:* todas las aplicaciones que procesan instancias RDCEO procesarán por lo menos el número de entradas indicadas. En otras palabras: una aplicación puede imponer un máximo de entradas que procesará para un elemento con múltiples valores, pero este máximo no debe ser menor que el valor máximo permitido mas pequeño.
- *Elementos de datos tipos CharacterString o LangString:* todas las aplicaciones que procesan instancias RDCEO procesarán por lo menos esa longitud para el valor CharacterString de un elemento de datos. En otras palabras: una aplicación puede imponer un máximo en el número de caracteres que procesarla para el valor CharacterString de un elemento, este valor no puede ser menor que el valor máximo permitido mas pequeño.

[Para descargar los ficheros del estandar originales pincha aquí](#)

14.12.4. Ver evaluación en Internet

14.13. IMS SIMPLE SECUENCING SPECIFICATION

14.13.1. Introducción

Este documento define de que manera y **en que orden** se deben presentar las diferentes actividades y piezas de contenido a los alumnos.

Se llama "simple" porque incluye un número limitado de comportamientos de secuenciación muy usados, no porque sea un estándar simple. Aunque este estándar no dicen nada de cómo tratar la secuenciación basada en inteligencia artificial o la que requiera datos o servicios externos (secuenciación de simulaciones embebidas).

Simple Sequencing permite presentar el contenido digital de un curso a un estudiante en un ambiente basado en la Web, aunque no sirve sólo para este ambiente. Sin embargo la comunicación, el control de los diferentes interfaces, así como los diferentes mecanismos para mediar las interacciones entre un estudiante y el sistema de aprendizaje tecnológico **LST (learning technology system)**, no forman parte de este estándar, así como tampoco forman parte del estándar la presentación, el estilo o los emplazamientos para el control de la navegación por la aplicación.



La Secuenciación Simple (Simple Sequencing) proporciona una representación externa, ampliando el "Paquete de Contenido IMS", para intercambiar descripciones de series o secuencias entre los diferentes componentes del LST. Hay que aclarar que aunque este estándar se basa en la misma organización en contenidos y la misma estructura que el "Paquete de Contenido", no le requiere.

14.13.2. Componentes de la Secuenciación Simple IMS (IMS Simple Sequencing Components)

La traducción hecha de este apartado del curso ha tenido en cuenta los siguientes documentos:

- **Simple Sequencing Information and Behavior Model**, este documento contiene la normativa a seguir para cumplir este estándar.
- **XML Binding**, describe como se presenta el Modelo de secuenciación como extensión de los componentes *<organization>* del manifiesto del Paquete de Contenido IMS.
- **Best Practice and Implementation Guide**, contiene las consideraciones a seguir para la implementación del estándar.

14.13.3. Secciones de la Especificación de la Secuenciación Simple (Simple Sequencing Specification Sections)

Esta especificación define un subconjunto de todas las posibles operaciones de seriación y de todos los elementos de información requeridos para describir todas estas operaciones deseadas, junto con sus parámetros correspondientes, en los cursos.

Esta especificación define, para cada curso, un subconjunto de todas las posibles operaciones de secuenciación y los elementos de información requeridos (junto con sus parámetros correspondientes) para describir todas estas operaciones deseadas. Para permitir interoperabilidad, los sistemas que distribuyan contenido de aprendizaje deben ser capaces de interpretar la información de secuenciación y mostrar el comportamiento definido por esta especificación.

Esta presentación esta dividida en diferentes secciones, cada sección de las que a continuación se menciona describe una parte del Simple Sequencing Specification y cada sección será vista en profundidad en posteriores apartado del capítulo que nos ocupa:



- **Modelo de Definición para la secuenciación** (Sequencing Definition Model), se usa para describir el comportamiento de la secuenciación. **(SM)**.
- **Modelo de Seguimiento** (Tracking Model), se usa para llevar la cuenta de los resultados obtenidos por los alumnos en sus interacciones con las diferentes actividades **(TM)**.
- **Modelo del Estado de la Actividad** (Activity State Model), se usa para recoger información sobre el estado de la interacción de un estudiante con una actividad en concreto (y los atributos globales de esta actividad). **(AM)**.
- **Proceso de Secuenciación Total o Global** (Overall Sequencing Process), es el proceso que relaciona los procesos de selección, aleatoriedad, navegación, secuenciación, salida y entrega **(OP)**.
- **Modelo de Comportamiento en la Navegación** (Navigation Behaviour Model), evalúa una petición de navegación y determina que solicitudes de secuenciación y terminación debe ser procesadas para identificar el contenido a enviar a un determinado alumno **(NB)**.
- **Modelo de Comportamiento de Finalización** (Termination Behaviour Model), evalúa una petición de finalización de una actividad **(TB)**.
- **Modelo de Comportamiento Enrollado** (Rollup Behaviour Model), calcula los resultados de una actividad a partir de los datos obtenidos en las actividades hijas de dicha actividad **(RB)**.
- **Modelo de Comportamiento de Selección y Aleatoriedad** (Selection and Randomization Behavior Model), selecciona y reordena un subconjunto de actividades **(SR)**.
- **Modelo de Comportamiento en Serie** (Sequencing Behaviour Model), evalúa una petición de secuenciación (de seguir con la secuencia) en los términos dados por el modelo de contenido del árbol de la actividad y determina que objeto debe ser enviado al alumno **(SB)**.
- **Modelo de Comportamiento de Entrega** (Delibery Behaviour Model), valida que se puedan enviar los recursos de una determinada actividad **(DB)**.
- **Procesos de Utilidad** (Utility Processes), procesos de secuenciación usados por múltiples modelos de comportamiento **(UP)**.



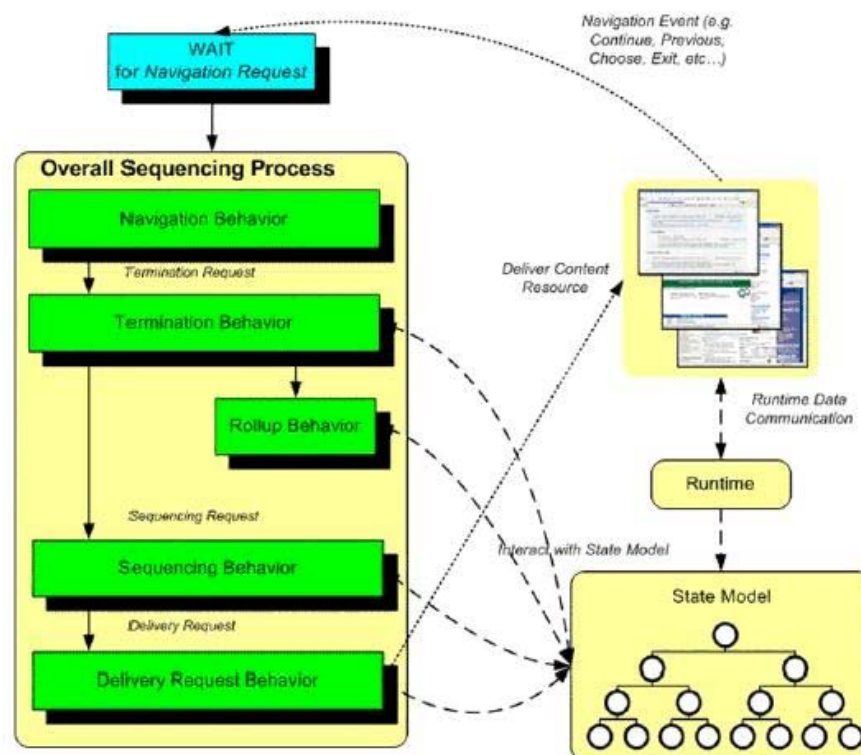
14.13.4. Suposiciones

1. **Árbol de actividad:** La Especificación de la Secuenciación Simple se basa en el concepto de **actividades de aprendizaje**, descritas en capítulos anteriores del curso. Por ello el contenido de la Secuenciación Simple está organizado en una estructura jerárquica de actividades y cada actividad debe incluir una o más actividades hijas. Cada actividad tiene asociada una lista de comportamientos en serie que describen cómo los hijos de una actividad se usan para crear una determinada experiencia de aprendizaje en cada caso.

El **SB** (Sequencing Behaviour Model), atraviesa el árbol de actividades en busca de las reglas de secuenciación para determinar que actividades y en que orden deben ser enviadas al estudiante para crear la experiencia de aprendizaje deseada. El estándar no especifica cómo crear, representar, y mantener el árbol de actividades y la definición de secuenciación asociada a él.

2. **Bucle de secuenciación:** Es un bucle que describe como los diferentes procesos de comportamiento interactúan durante la secuenciación y el envío de la secuencia a seguir:
 - Comienzo de la sesión:
 1. El alumno accede al sistema y selecciona un curso.
 2. El LTS inicia una sesión de secuenciación permitiendo una petición de navegación "Start", "Resume All" o "Choice".
 3. El comportamiento de navegación procesa esta petición.
 - Comienzo de la secuenciación:
 1. Usando la información del Modelo de Seguimiento TM se recorre el árbol actividades para tratar de identificar a la actividad adecuada que será enviada al estudiante.
 2. El comportamiento de entrega, determina si esa actividad se puede enviar, y si es así, la prepara para ello. Sino se para.
 3. El estudiante interactúa con el recurso dado. El proceso de secuenciación espera peticiones mientras tanto.
 4. La actividad manda valores que actualicen los elementos del modelo de seguimiento durante el paso anterior.
 5. El estudiante o la actividad invoca un evento de navegación como "Previous", "Exit", etc.

6. El LTS informa sobre el proceso de secuenciación del evento mediante una petición de navegación.
 7. El comportamiento de navegación transforma esta petición a una de secuenciación que hace que si dicha petición es de salida, la sesión termine.
 8. Si se sale se debe actualizar antes el modelo de seguimiento. El proceso enrollado actualizará también el modelo para los padres de la actividad correspondiente.
 9. El proceso se repite empezando en el apartado 1. hasta que se cierre la sesión.
- Este proceso se ve en la siguiente imagen:



3. **Modelo Basado en Reglas.** Las actividades llevan asociadas una información de secuenciación en el árbol de actividades que incluye una serie de reglas para el comportamiento condicional.
4. **Comportamiento por defecto y autorizado:** En ausencia de información de secuenciación específica, los árboles de actividades tienen un comportamiento por defecto especificado en la información de secuenciación, ver **SM**.



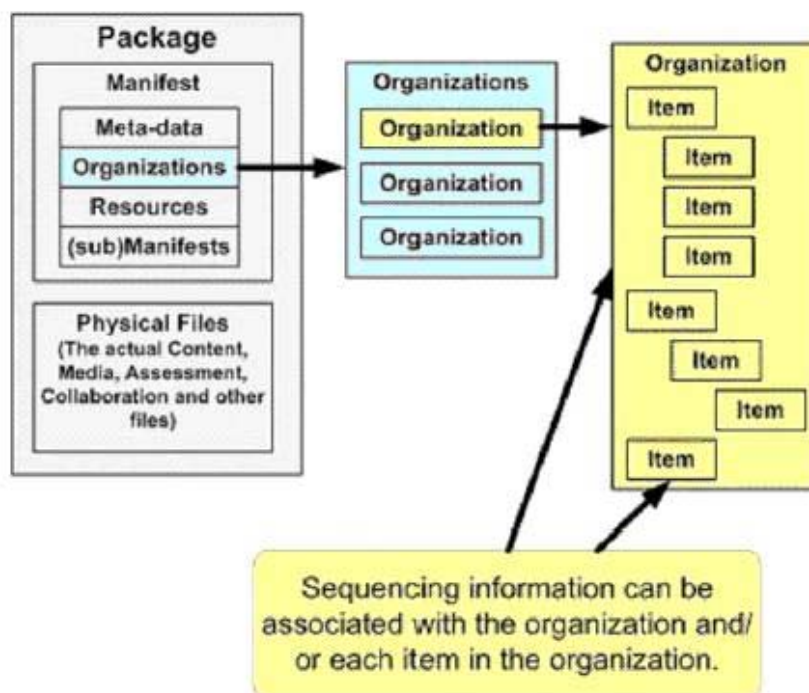
5. **Objetivos:** Los objetivos de aprendizaje están separados de las actividades. Estos primeros representan un conjunto de ítems de datos locales o globales a los que se quiere llegar, cada uno con un estado de satisfacción diferente. La Secuenciación Simple no hace ninguna asunción sobre como interpretar dichos objetivos. Las actividades pueden tener más de un objetivo local asociado y pueden referenciar a varios objetivos globales compartidos. La resolución de los objetivos locales y globales ID no esta especificada en este estándar.
6. **Recursos auxiliares:** Una actividad puede tener asociada recursos auxiliares que den servicios y recursos adicionales al estudiante. Pero esta especificación no dice nada sobre ellos, sólo da un medio por el que estos recursos pueden estar asociados a una actividad.
7. **Presentación y Ambiente:** Este estándar no hace ninguna asunción sobre como los contenidos o controles deben ser prestados o presentados al estudiante (lugar, GUI, estilo...).
8. **Actividades suspendidas y reanudadas:** Una actividad puede ser suspendida y luego reanudada o puede ser abandonada tras ser suspendida. Habrá actividades que puedan llevar a suspender otras, etc...
9. **Comenzar y detener el proceso de secuenciación:** Este estándar no dice como empezar o parar el proceso de secuenciación total. Normalmente será mediante un login y logout.
10. **Persistencia de los datos:** El estándar tampoco especifica como los datos de seguimiento deben ser persistentes a lo largo de múltiples instancias de un proceso de secuenciación de un estudiante y un árbol de actividades a través de múltiples sesiones con un login. La implementación debe ser persistente con el control, el seguimiento y los datos, por lo menos hasta que el intento actual sobre el árbol de actividades haya terminado. Eso puede incluir una o mas sesiones.
11. **Tipos de contenido:** Esta especificación es independiente de los tipos del contenido de aprendizaje y los objetivos de aprendizaje y puede ser usada para seriar o secuenciar cualquier clase de contenido: archivos DOC, PDF..., Paginas Web estáticas, recursos MIME...
12. **Contenido Activo/Pasivo:** La Secuenciación Simple recae en los valores del modelo de seguimiento para controlar la forma de hacer las secuencias. Este estándar diferencia el contenido activo (pone directamente elementos en el modelo de seguimiento) del pasivo (tiene unos valores ya establecidos por defecto) y soporta ambos tipos.



13. Relación con la Especificación del Paquete de Contenido IMS:

Como ya se ha mencionado, la Especificación de la Secuenciación Simple (Simple Sequencing Specification) recae en el concepto de **actividades de aprendizaje**, y la Especificación del Paquete de Contenido IMS da una estructura predefinida para relacionar la actividad de aprendizaje con el recurso de contenido- el elemento *<item>* y su relación con el elemento *<resource>*. Además, los elementos *<item>* se pueden agrupar en colecciones que serán contenidas en un elemento *<organization>*, La Especificación del Paquete de Contenido se extiende con la información de secuenciación que define como esta información se asocia al paquete de contenido. El proceso de definir una secuencia específica de actividades de aprendizaje empieza con la creación de un contenido de agregación, ya que las agregaciones se pueden intercambiar entre sistemas. Como se ve en la figura que aparece a continuación, el elemento *<organization>* del paquete de contenido y cada elemento *<item>* dentro de él, pueden haber definido comportamientos de secuenciación a través de la asociación de información de secuenciación definida en el Modelo de definición de secuenciación (**SM**). Se pueden crear y referenciar diferentes conjuntos de definiciones de secuenciación para ítems dentro de una organización.

14. Toda la información que define el comportamiento de secuenciación para una organización será presentada dentro de la propia organización. Cuando varias organizaciones aparecen en un manifiesto, cada organización puede tener su propio comportamiento de secuenciación. Este estándar no prohíbe que la información de secuenciación y el comportamiento se usen en otros contextos. Además esta versión no prohíbe el uso de sub manifiestos. Ver imagen:



14.13.5. SM. Modelo de definición de la secuenciación (Sequencing Definition Model)

El proceso de Secuenciación Simple usa información sobre el comportamiento de diferentes secuencias deseado para controlar la secuenciación, selección y distribución de las actividades al estudiante. Para ellos se describe la secuencia a seguir por las actividades del curso mediante un conjunto de atributos. Estos atributos se asocian con las actividades de aprendizaje en el árbol de actividades para describir el comportamiento a seguir por la secuenciación.

El conjunto de atributos usados por la Secuenciación Simple se llama "modelo de definición de la secuenciación" (sequencing definition model), el cual consiste en:

- **SM.1: Modos de Control de la Secuenciación (Sequencing Control Modes):** controles para los diferentes tipos de peticiones de secuenciación que se pueden aplicar a una colección de actividades. Son opcionales. Y si no aparecen se usarán sus valores por defecto. Los controles son: Choice, Choice Exit, Flow, Forward Only, Objective Information, Progress Information. Para ver una descripción completa de los elementos, pincha [aquí](#).



- **SM.2: Descripción de la Norma o Regla de secuenciación (Sequencing Rule Description):** son el conjunto de atributos mostrados en la tabla que se adjunta al final y que se aplican a una actividad para especificar los distintos comportamientos de secuenciación para dicha actividad. Son opcionales. Cada actividad puede tener un número ilimitado de reglas de este tipo, que se definen sólo cuando se necesitan.

La forma general de una regla es: *If condition set Then action*. Donde puede haber múltiples condiciones combinadas por un and (y) o un or (o) y además las condiciones simples pueden ser también negadas antes de combinarse. Cada condición referencia un ítem en el modelo de seguimiento de la actividad y debe de haber una acción (action) que se de como resultado si las condiciones evalúan a cierto (true).

Las acciones se dividen en tres grupos: Precondiciones (preconditions) se incorporan en reglas que deciden si una actividad se distribuye o no, Poscondiciones (Post Conditions) se incluyen en reglas que se apliquen cuando una actividad termina y Acciones de Salida (Exit Action) se incluyen en reglas que se aplican cuando existe un descendiente de una actividad. Para ver una descripción completa de los atributos, [pincha aquí.](#)

- **SM.3: Descripción de Condiciones Límite (Limit Conditions Description):** límites sobre cuántas veces, cuánto tiempo y cuando se permite una actividad. Son opcionales, por lo que no necesitan ser definidas para todas las actividades del árbol de actividades. Para ver todo el conjunto de atributos, [pincha aquí.](#)
- **SM.4: Descripción del Recurso Auxiliar (Auxiliary Resource Description):** define un recurso auxiliar asociado a una actividad, así cuando se envía la actividad al estudiante, se le envía también el recurso. Los recursos auxiliares son opcionales. Los atributos que se listan en la siguiente tabla describen un recurso auxiliar, y cada actividad puede tener un número ilimitado de recursos auxiliares sin ordenar. [Pincha aquí.](#)
- **SM.5: Descripción de la Norma Enrollada (Rollup Rule Description):** reglas que especifican como los datos de seguimiento de una actividad o los objetivos asociados a esa actividad son producidos a partir de los resultados obtenidos de las actividades hijas. Es decir, describen como los valores de las actividades hijas influyen en la Información de Progreso de una actividad. Son opcionales y se definen sólo donde se necesitan. Los atributos de la tabla adjunta describen una norma y cada actividad puede tener un número ilimitado de reglas sin ordenar.



El formato general de una regla puede ser: *If child-activity set, condition set Then action*. Donde el Child activity set, dice que actividades hijas se requieren para que la condición sea cierta y donde se permiten varias condiciones que pueden estar unidas por un and o un or, y también pueden ser negadas. Para ver los atributos [pincha aquí](#).

- **SM.6: Descripción del Objetivo:** objetivos de aprendizajes asociados a una actividad. Cada actividad puede tener un número ilimitado de objetivos de aprendizaje y el modelo de seguimiento (**TM**) define un conjunto de datos que guardan el estado de satisfacción (suspendido, aprobado...) o el estado de medida (puntuación...) de cada objetivo. El significado de cada objetivo de aprendizaje no está definido en este modelo, sólo está definido en términos de su ID y su asociación con una actividad. Una actividad tendrá información objetiva para cada objetivo asociado a dicha actividad. Y una actividad debe tener **al menos un** objetivo. Además una actividad tendrá uno y sólo un objetivo que contribuya al enrollado de actividades padres e hijas. El objetivo ID, es necesario si la actividad tienen mas de un objetivo o si la información del objetivo de la actividad se comparte con otras actividades. [Ver tabla](#).
- **SM.7: Mapa del Objetivo:** la localización de la información objetiva local de y para los objetivos globales compartidos. Cada actividad puede tener un número ilimitado de mapas de objetivo. Por defecto no se comparte información de objetivos entre las actividades. Para ver la tabla [pincha aquí](#).
- **SM.8: Controles para el Enrollado:** controles y datos de una actividad para poder ascender a su actividad antecesora o padre. Incluye los tipos de "comportamiento para enrollados" específicos para una actividad. No es necesario definirlos para cada actividad del árbol de actividades. [Ver tabla con todos los atributos](#).
- **SM.9: Controles de Selección (Selection Controls):** controlan como las actividades hijo de la actual se seleccionan durante el proceso de secuenciación. Los procesos de secuenciación Simple deben referenciar los datos de control de selección para toda actividad del árbol de actividades. Son opcionales. [Para ver todos los atributos](#).
- **SM.10: Controles de aleatoriedad (Randomization Controls):** controlan como las actividades hijo de la actual se ordenan. Los procesos de secuenciación Simple deben referenciar los datos de control de aleatoriedad para toda actividad del árbol de actividades. Son opcionales. [Ver tabla](#).



- **SM.11: Controles de distribución (Delivery Controls):** para controlar cuando se guarda la información objetiva y el progreso en una actividad. Describen acciones y controles usados cuando se envía o distribuye una actividad, es decir, Objetivo, Actividad y Datos del Progreso se recogen cuando se distribuye la actividad. Los procesos de secuenciación Simple deben referenciar los datos de controles de distribución para toda actividad del árbol de actividades. Son opcionales. [Ver tabla de atributos.](#)
- **SM.12: Descripción de la Secuenciación (Sequencing Description):** el conjunto completo de información de secuenciación (todos los ítems descritos arriba) está asociado a cada actividad en el árbol de actividades.

[La tabla adjunta](#) lista todos los elementos de toda la descripción de la secuenciación y sus multiplicidades.

El mecanismo que se use para crear o mantener la información del modelo de definición de datos para la secuenciación no forma parte de este modelo.

Una implementación debe ser capaz de representar todo el rango de valores descritos hasta el momento.

14.13.6. TM. Modelo de Seguimiento (Tracking Model)

Los procesos de Secuenciación Simple usan información sobre los resultados de las interacciones de los estudiantes con las actividades y los objetivos conseguidos por éstos para controlar la selección y secuencia que siguen otras actividades.

Los comportamientos de secuenciación se definen mediante un conjunto limitado de atributos de datos específicos que describen los resultados de las interacciones de los estudiantes. El conjunto de atributos definido y usado por la Secuenciación Simple se dice que es el "Modelo de Seguimiento".

El Modelo de Seguimiento consiste en dos partes:

- TM.1: Modelo de Información de Seguimiento.
- TM.2: Comportamientos del Modelo de Seguimiento.

Cada una de las partes del modelo, arriba mencionadas, se detallan a continuación:

- **TM.1: Modelo de Información de Seguimiento**, describe los datos usados por un sistema que distribuye actividades en serie, pero no dice como se codifican, almacenan o representan dichos datos, está definido en dos partes:



- **TM.1.1: Información del progreso del objetivo:** información sobre los resultados del estudiante relacionados con el objetivo que debe cumplir para una determinada actividad. La información de seguimiento para un objetivo incluye los datos asociados a ese objetivo (que se listan [en la tabla asociada](#)), es decir, la información del objetivo, pero no se indica como esa información se asocia al objetivo.
- **TM.1.2: Información del progreso de la Actividad / Intento:** Información sobre los diferentes intentos del estudiante en la actividad.
 - **TM.1.2.1: Información del progreso de la Actividad:** formada por el conjunto de atributos [de la tabla asociada](#), describe los progresos de un alumno en una actividad en concreto. No se define en este modelo, sin embargo, el mecanismo para definir la duración de una actividad, ni como la información de progreso se asocia a esa actividad.
 - **TM.1.2.2: Información del progreso del Intento:** compuesta por el conjunto de atributos de [la tabla asociada](#), describe el progreso de un estudiante para un único intento en una actividad individual. En el modelo no se especifica como esta información se asocia a la actividad correspondiente.
- **TM.2: Comportamientos del Modelo de Seguimiento:** La información del progreso de un objetivo se aplica sólo a dicho objetivo por lo que debe ser instanciada para cada objetivo ya sea este local o global (compartido de cada estudiante) y para cada nuevo intento que haga el alumno en cada actividad.

La información del progreso de la actividad es para una única actividad por lo que se instancia para cada actividad del estudiante.

La información del progreso del intento es para un intento. Será instanciado para cada estudiante y debe ser inicializado para cada nuevo intento en cada actividad.

LTS debe mantener las duraciones de Actividad e Intento (absolutas y experimentadas). En este documento no se definen los mecanismos o procesos para gestionar las duraciones. Sólo se definen:

- La Duración Absoluta de la Actividad (TM.1.2.1) será la suma de todas las Duraciones Absolutas de los Intentos de la actividad.



- La Duración Absoluta de la Actividad (TM.1.2.1) de una actividad padre será la suma de todas las Duraciones Absolutas de las Actividades de sus hijos. [. Ver la siguiente figura \(TM.1\).](#)

La duración absoluta se cuenta desde que empieza la actividad hasta que termina. La duración experimentada se acumula desde que empieza cada intento hasta que se suspende.

Para referenciar un objetivo específico se requieren el ID del objetivo y un identificador de actividad asociado. Si la actividad sólo tiene un objetivo local y no está compartido con otra actividad, sólo se necesita el identificador de la actividad.

Cuando está indefinido un dato del objetivo local (el "Objective Progress Status" (TM.1.1) o el "Objective Measure Status" (**TM.1.1**) son falsos) y está asociado con el objetivo un 'read' "Objective Map" (**SM.7**), se usa la correspondiente información del objetivo global compartido. Esto no afecta a la información local del objetivo. La referencia al objetivo global solo usa el ID del objetivo para localizar la información real del objetivo.

Los procesos de Secuenciación Simple acceden a la información del objetivo local y del progreso del intento sólo para el intento más reciente. Si el alumno está interactuando con la actividad, el intento más reciente es la interacción o el intento actual. De otro modo, el intento más reciente es el último completado.

Los elementos "UseCurrent Attempt Objective Information" (**SM.1**) y Use Current Attempt Progress Information (**SM.1**) controlan el acceso a la información de seguimiento para el intento más reciente en una actividad. Si cualquiera de los dos es "Cierto" para una actividad, el modelo de seguimiento no mantendrá los datos de seguimiento del intento más reciente para ningún hijo de la actividad, a menos que estos ocurran mientras esté activo el intento actual de su actividad padre.

No hay requisitos sobre cómo o cuando los valores de los datos de seguimiento se establecen o actualizan para el alumno. La Secuenciación Simple sólo se fija en los valores 'actuales' en el modelo de información. Generalmente las interacciones del estudiante con una actividad darán como resultado los cambios apropiados de los datos de seguimiento, pero no se especifican estos mecanismos.

Los procesos de finalización, secuenciación y entrega (ver TB, SB y DB) usan los datos de seguimiento para que los elementos del árbol de actividad determinen que sucede cuando un intento sobre una actividad termina y cómo las actividades posteriores son secuenciadas y entregadas al alumno.

No se especifican como parte del modelo de información los mecanismos usados para guardar y establecer los datos de seguimiento.

14.13.7. AM. Modelo del Estado de la Actividad (Activity State Model)

Los procesos de Secuenciación Simple usan información sobre el estado de las interacciones del estudiante con las actividades. El conjunto definido de los atributos de estado usados por Simple Sequencing se llama el "modelo de estado de actividad".



Este modelo consiste en:

- AM.1: Modelo de información de Estado de Actividad.
- AM.2: Comportamientos del Modelo de Estado de Actividad.

Cada una de las partes del modelo, arriba mencionadas, se detallan a continuación:

- **AM.1: Modelo de información de Estado de Actividad:** describe los datos usados por un sistema que entrega actividades secuenciadas. Esta especificación no dice nada sobre como se codifica, almacena, representa, etc. esta información. El modelo de información de estado sólo describe un conjunto de ítems de datos relacionados y obligaciones internas sobre los valores de dichos ítems. Se define en dos partes:
 - **AM.1.1: Información de Estado de Actividad.** Es un conjunto de atributos ([Ver tabla](#)) que describen el estado de una actividad de un alumno. Los datos de estado de actividad deben ser instanciados para cada actividad en el árbol de actividades para cada alumno. Como se asocia esta información con una actividad no se especifica.
 - **AM.1.2: Información de Estado Global.** Es un conjunto de atributos ([Ver tabla](#)) que describen el estado de un alumno en la experiencia de aprendizaje secuenciada global. Esta información debe instanciarse una vez para un árbol de actividad para cada alumno. Cómo se asocia la información con un árbol de actividad no se especifica.
- **AM.2: Comportamientos del Modelo de Estado de Actividad.** El dato de estado de actividad es para una actividad y debe instanciarse para cada actividad para cada alumno.

El dato de estado global es para el proceso de secuenciación total y debe ser instanciado para el proceso de secuenciación entero para cada alumno.

No se especifican los mecanismos usados para grabar y establecer los datos de estado de actividad.

Aunque el comportamiento detallado para cambiar la Actividad Actual se describe en los procesos de finalización, secuenciación y entrega que se explican después, el [diagrama de transición asociado](#) es un resumen de los comportamientos.



14.13.8. OP. Proceso de Secuenciación Total o Global **(Overall Sequencing Process)**

Combina todos los elementos de la Secuenciación Simple para producir la interacción y relaciona además todos los procesos y comportamientos:

- Modelo de Comportamiento en la Navegación
- Modelo de Comportamiento de Finalización
- Modelo de Comportamiento de Selección y Aleatoriedad
- Modelo de Comportamiento en Serie
- Modelo de Comportamiento de Entrega

El proceso de secuenciación total describe sólo un flujo de trabajo lógico, no asume nada sobre cómo ni cuando se invocan los pasos del proceso o en qué orden se procesan.

OP.1: Comportamiento Total (Overall Behaviour).

Cuando el alumno inicia el proceso, el LTS (Learning Technology System) genera una petición de navegación "Start" o "Resume All". Otras interacciones generarán sus peticiones pero no se especifica cómo o cuando se generan estas peticiones.

El sistema opera en un bucle hasta que el LTS lo termina, esperando peticiones de navegación externas, procesándolas, posiblemente entregando el contenido y entonces espera más peticiones.

Las peticiones de navegación se transforman en peticiones de finalización y secuenciación. El proceso de finalización termina la actividad con la petición de finalización (sólo una actividad puede estar activa). Las reglas del proceso de terminación pueden ignorar la petición de secuenciación. El proceso de enrollado propaga los resultados desde una actividad a sus ancestros. Se procesa la petición de secuenciación para determinar la próxima actividad. Si se identifica una, se pasa al proceso de entrega, que determina si puede ser entregada (es decir, cumple las limitaciones y reglas). Al encontrar una actividad para entregar, el LTS da la descripción de los recursos auxiliares y el contenido de los recursos asociados a una actividad se presenta al estudiante. Una vez presentado el contenido al alumno, el "Proceso de Secuenciación Total" espera otra petición de navegación.

Una sesión puede lanzarse de una de estas tres formas:

1. Una petición de secuenciación "Start".
2. Una petición de secuenciación "Resume All".
3. Una petición de secuenciación "Choice" antes de cualquiera de las dos anteriores.



Una sesión de secuenciación termina normalmente cuando se procesa una petición de secuenciación "Exit" desde la raíz del árbol de actividad.

El comportamiento total o global asume que una petición se procesa totalmente antes de procesar otra. El Proceso de Secuenciación Total está especificado [por el siguiente pseudo-código](#). Describe sólo el procesamiento, no se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.

14.13.9. NB. Modelo de Comportamiento en la Navegación (Navigation Behaviour Model)

Cuando un estudiante interactúa con un interfaz, cada petición para moverse por el contenido genera un "evento de navegación", por ejemplo, un clic para ir a la siguiente actividad. Cada petición de navegación lanza una petición de finalización y de secuenciación que serán usadas para terminar el intento de la actividad actual y determinar la próxima actividad. El proceso de evaluar la petición de navegación e identificar las peticiones de finalización y secuenciación apropiadas (o devolver un error) se dice que es el "proceso de navegación".

No asume nada sobre cómo o cuando se genera una petición de navegación, ni sobre los interfaces de usuario. Tampoco lo hace sobre cómo o cuando las peticiones de finalización y secuenciación identificadas serán procesadas. Las peticiones, en la mayoría de los casos, se pasarán inmediatamente por el proceso de secuenciación total (**OP**) a los procesos de finalización y secuenciación.

El proceso de navegación sólo usa datos "Sequencing Control Mode" del modelo de definición de secuenciación (**SM.1**).

El proceso de navegación no usa datos del modelo de seguimiento (**TM**), pero si del modelo de estado de actividad (**AM**).

El comportamiento del proceso de navegación se define en términos de un proceso único: Proceso de Petición de Navegación.

- **NB.1: Peticiones de Navegación (Navigation Requests):** El proceso de navegación responde a una petición de un [conjunto de peticiones dadas](#). Una petición define las acciones a realizar cuando es procesada.
- **NB.2: Comportamiento de Navegación (Navigation Behaviour):** Describe cómo un procesador de navegación interpreta una petición e identifica las correspondientes peticiones de finalización y secuenciación.



- **NB.2.1: Proceso de Petición de Navegación (Navigation Request Process):** Examina la petición de navegación, la valida y determina las correspondientes peticiones de finalización y secuenciación. Este proceso está especificado [especificado por el siguiente pseudo-código](#). Describe sólo el procesamiento y conversión a peticiones de finalización y secuenciación, no se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.

14.13.10. TB. Modelo de Comportamiento de Finalización (Termination Behaviour Model)

Las actividades terminan debido a otros sucesos y peticiones diferentes a los vistos hasta ahora. Cuando una actividad termina, el suceso lanzado puede implicar que una o más actividades finalicen. El proceso de evaluar la acción de salida y post condiciones, guardando información sobre el estado de la actividad cuando termina, y finalizar otras actividades basándose en los comportamientos definidos es lo que se llama "proceso de finalización".

Este proceso no asume nada sobre cómo o cuando se genera la petición de finalización.

El proceso de finalización está controlado por partes del modelo de definición para la secuenciación (Definiciones de Reglas de Secuenciación), usa partes del modelo de seguimiento ("Objective Information" y "Progress Information") y usa datos del modelo de estado de la actividad ("Activity State Information"). El comportamiento del proceso se define por tres procesos (Proceso de Petición de Finalización, Proceso de Fin de Intento (**UP.4**) y "Proceso de Enrollado Total") y dos subprocesos ("Subproceso de Reglas de Acción de Salida" y "Subproceso de Reglas de Post Condición").

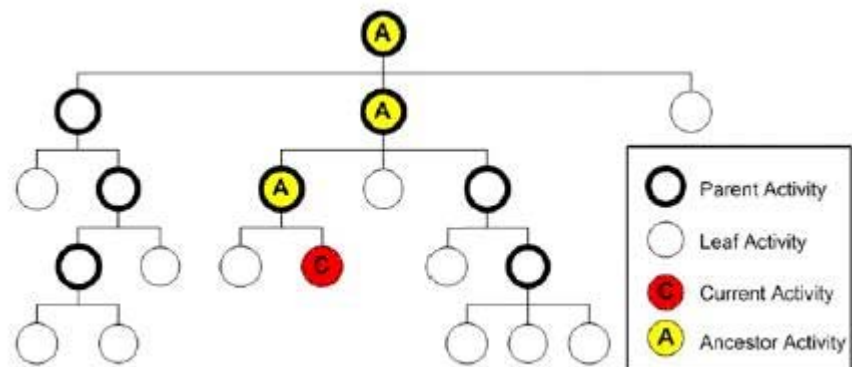
- **TB.1 Peticiones de Finalización (Termination Request):** El proceso de finalización responde a una de [un conjunto de peticiones de finalización](#). Una petición define las acciones a realizar cuando es procesada.
- **TB.2 Comportamiento de Finalización (Termination Behaviour):** Describe cómo se procesa una petición de terminación. Marca como terminados los intentos de las actividades, actualiza el estado del árbol de actividad, procesa las reglas de acción de salida y post condiciones, y podría identificar una petición alternativa.

El comportamiento de finalización depende de las descripciones de datos de los modelos de definición de secuenciación (**SM**), de seguimiento (**TM**) y de estado de actividad (**AM**).

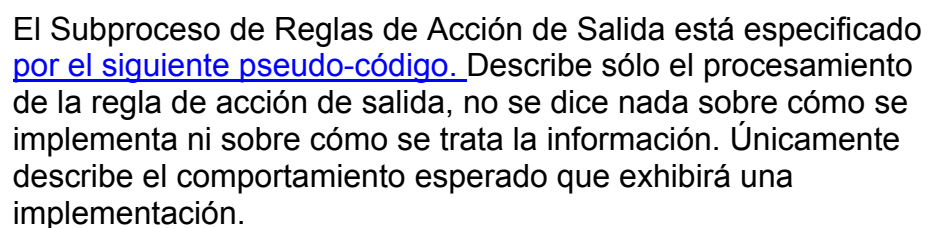
- **TB.2.1: Subproceso de Reglas de Acción de Salida (Sequencing Exit Action Rules Subprocess):** Las reglas de acción de salida sobre los antecesores de la actividad actual se evalúan cuando termina el intento de la actividad actual.

Las condiciones de las reglas se especifican en las "Definiciones de Reglas de Secuenciación" y se evalúan usando el "Proceso de Comprobación de Reglas" (**UP.2**).

Este subproceso asume que se ha completado el "Proceso de Enrollado Total o Global" (**RB.1.4**). Evalúa sólo las reglas cuyas acciones son reglas de Acción de Salida ("Exit Action", ver **SM.1.2**) para todos los ancestros de la actividad actual. Se evalúan en orden empezando por la raíz. El subproceso para en la primera regla que vale Cierto. Si un ancestro termina, todos sus descendientes activos también finalizan y esa actividad se marca la actividad actual. Ver el siguiente ejemplo:



La actividad actual y todos sus ancestros están activos. La actividad actual inicia el proceso de petición de Salida <Exit", por ejemplo, una petición de navegación de salida. Se aplica el Proceso de Fin de Intento (**UP.4**). Cuando se aplica el "Subproceso de Reglas de Acción de Salida", suponemos que termina un ancestro, el más cercano a la raíz, y tiene a "Regla de Acción de Salida" que vale "Cierto". Asumimos que las actividades madre y abuela de la actual tienen Reglas de Acción de Salida ciertas, sólo terminará la actividad abuela. Mediante este subproceso se termina el intento del ancestro que finaliza y de todos sus descendientes activos aplicando el "Proceso de Intentos de Descendientes Finalizado" (**UP.3**) y marca como actividad actual la identificada que terminaba.



- Las reglas se evalúan según el Proceso de Evaluación de Reglas (UP.2).

Este subproceso asume que se ha completado el "Subproceso de Reglas de Acción de Salida". Evalúa sólo las reglas cuyas acciones están definidas como reglas de "Post Condición" (**ver SM.1.2**) sobre la actividad actual. Opcionalmente puede devolver una y sólo una petición de finalización y una y sólo una petición de secuenciación, determinadas por la acción de la regla.

El Subproceso de Reglas de Post Condición está especificado [por el siguiente pseudo-código](#). Describe sólo el procesamiento de la regla de post condición, no se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.



- **TB.2.3: Proceso de Petición de Finalización (Termination Request Process).** Usa el Proceso de Enrollado Total (**RB.1.4**), Subproceso de Reglas de Acción de Salida (**TB.2.1**), Subproceso de Reglas de Post Condición (**TB.2.2**) y el Proceso de Fin de Intento (**UP.4**). Actualiza valores en el modelo de seguimiento y de estado de actividad.

Termina la actividad actual y actualiza la información del estado del árbol de actividades. El resultado del "Proceso de Petición de Finalización" puede identificar una nueva petición de secuenciación.

Ciertas peticiones de finalización ("Suspend All", "Abandon" y "Abandon All") no cambiarán el estado del modelo de seguimiento de la actividad actual.

Proceso de Petición de Finalización está especificado [por el siguiente pseudo-código](#). Describe sólo el procesamiento de la petición de finalización, no se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.

14.13.11. RB. Modelo de Comportamiento de Enrollado (Rollup Behaviour Model)

El proceso de computar los resultados de una actividad a partir de los resultados de sus hijos se llama el "proceso de enrollado" o sólo "enrollado".

El proceso de enrollado se usa para mantener los datos del modelo de seguimiento para todas las actividades y objetivos del árbol de actividades.

El proceso de enrollado se aplica a todas las partes del modelo de seguimiento (**TM**) y está controlado por partes del modelo de definición de secuenciación (**SM**): "Controles de Enrollado", "Definiciones de Reglas de Enrollado", "Descripción de Objetivo" y "Controles de Entrega".

Está compuesto por las siguientes secciones:

- **RB.1: Comportamiento de Enrollado (Rollup Behaviour).**
 - RB.1.1: Proceso de Medida de Enrollado (Measure Rollup Process).
 - RB1.2: Proceso de Enrollado de Objetivo (Objective Rollup Process).
 - RB1.3: Proceso de Progreso de Actividad de Enrollado (Activity Process Rollup Process).



- RB1.4: Subproceso de Evaluación de Reglas de Enrollado (Rollup Rule Check Subproceso).
- RB1.5: Proceso de Enrollado Total (Overall Rollup Process).

Cada uno de ellos se explica a continuación en profundidad.

RB.1: Comportamiento de Enrollado (Rollup Behaviour). Describe cómo interpreta un sistema los elementos del modelo de definición de secuenciación en combinación con los datos de instancia del modelo de seguimiento para actualizarle.

El comportamiento de enrollado depende de los datos de los modelos de definición de secuenciación (**SM**) y de seguimiento (**TM**).

- **RB.1.1: Proceso de Medida de Enrollado (Measure Rollup Process).** Se usa para determinar la medida normalizada para el objetivo asociado a una actividad y cuyo valor del atributo "Objective Contribute to Rollup" (**SM.6**) es "Cierto". Se calcula a partir de las medidas de los objetivos asociados a los hijos que tienen el mismo atributo con valor "Cierto".

Este proceso se define por: Se incluyen en el cómputo todos los hijos seguidos con un "Rollup Objective Measure Weight" (**SM.8**) no negativo. Si es Falso el "Objective Measure Status" (**TM.1.1**) de una actividad hija con el atributo "Objective Contributes to Rollup" (**SM.6**) vale Cierto, entonces es Falso el "Objective Measure Status" para la actividad. De otro modo, calcula el peso medio de todos los hijos. El cómputo usa los objetivos de las actividades hijas que tienen el atributo "Objective Contributes to Rollup" que vale Cierto.

La suma de los valores de "Rollup Objective Measure Weight" para los hijos no necesita ser 1.0.

El "Proceso de Medida de Enrollado" está especificado [por el siguiente pseudo-código](#). Describe sólo la lógica de medida de enrollado, no se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.

- **RB1.2: Proceso de Enrollado de Objetivo (Objective Rollup Process).** Se usa para determinar el estado del objetivo asociado a una actividad a partir de los valores de los hijos.



Este proceso se define por:

1. 1) *Cómputo Basado-en-Medida*: Se satisface el objetivo si la "Medida Normalizada de Objetivo" (**TM.1.1**) es igual o mayor que la "Medida Normalizada Mínima para Satisfacer al Objetivo" (**SM.6**). Se usa el enrollado basado-en-medida si y solo si el atributo "Objetivo Satisfecho por Medida" (**SM.6**) es Cierto para el objetivo de la actividad que tiene el atributo "Objetivo Contribuye al Enrollado" que vale Cierto.
2. 2) *Cómputo Basado-en-Reglas*: Se evalúan las reglas de enrollado con "Acciones de Enrollado" (**SM.5**) satisfechas (satisfied) o no satisfechas. Para evaluar las "Condiciones de Enrollado" (**SM.6**) de la regla se usan las actividades hijas, que tienen los atributos "Objetivo de Enrollado Satisfecho" (**SM.8**) y "Seguido" (**SM.11, Tracked**) iguales a Cierto. Si la regla de enrollado evalúa a cierto, la "Acción de Enrollado" pone valores al "Estado de Progreso del Objetivo" y al "Estado Satisfecho del Objetivo". Sólo se hace el cómputo si el objetivo tiene el atributo "Objetivo Contribuye al Enrollado" igual a Cierto. Primero se evalúan las reglas con "Acción de Enrollado no satisfecho (Not Satisfied)" y después las satisfechas, así estas no hacen caso de la reglas no satisfechas.
3. 3) *Cómputo por defecto Basado-en-Reglas*: la actividad no tiene definidas reglas de enrollado cuyas "Acciones de Enrollado" sean Satisfechas o No Satisfechas, el cómputo basado-en-reglas descrito arriba se aplica usando las siguientes reglas de enrollado por defecto:
 - Regla por defecto *Satisfecha*: "Conjunto de Hijos de Enrollado" valor de "Todos" (All), "Condición de Enrollado" valor de "Satisfecho", y "Acción de Enrollado" valor de "Satisfecho".
 - Regla por defecto *No Satisfecha*: "Conjunto de Hijos de Enrollado" valor de "Todos" (All), "Condición de Enrollado" valor de "No Satisfecho" y (la Condición de Combinación de Todo) Intentado, y "Acción de Enrollado" valor de "No Satisfecho".

Proceso de Enrollado de Objetivo está especificado [por el siguiente pseudo-código](#). Describe sólo la lógica de enrollado de objetivo, no se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.



- **RB1.3: Proceso de Progreso de Actividad de Enrollado (Activity Process Rollup Process).** Se usa para determinar el estado de completitud del intento actual a partir de los hijos de la actividad. El "comportamiento de enrollado" del progreso de actividad está definido por:
 1. Cómputo Basado-en-Reglas: Se evalúan las reglas de enrollado con "Acciones de Enrollado" (**SM.5**) incompletas (Incomplete) o completas. Para evaluar las "Condiciones de Enrollado" (**SM.6**) de la regla se usan las actividades hijas, que tienen los atributos "Progreso de Enrollado Completo y Seguido" (**SM.11: Tracked**) iguales a Cierto. Si la regla de enrollado evalúa a cierto, la "Acción de Enrollado" pone valores al "Estado de Progreso del Intento" y al "Estado de Completitud del Intento" al intento actual de la actividad. Primero se evalúan las reglas con "Acción de Enrollado" incompleto (Incomplete) y después las completadas, así estas no hacen caso de la reglas incompletas.
 2. Cómputo por defecto Basado-en-Reglas: Si la actividad no tiene definidas reglas de enrollado cuyas "Acciones de Enrollado" sean Incompletas o Completas, el cómputo basado-en-reglas descrito arriba se aplica usando las siguientes reglas por defecto:
 - Regla por defecto *Completa*: "Conjunto de Hijos de Enrollado" valor de "Todos" (All), "Condición de Enrollado" valor de "Completa", y "Acción de Enrollado" valor de "Completa".
 - Regla por defecto *Incompleta*: "Conjunto de Hijos de Enrollado" valor de "Todos" (All), "Condición de Enrollado" valor de "Incompleta" y (la Condición de Combinación de Todo) Intentado, y "Acción de Enrollado" valor de "Incompleta".

El "Proceso de Progreso de Actividad de Enrollado" está especificado [por el siguiente pseudo-código](#). Describe sólo la lógica de progreso de actividad de enrollado, no se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.



- **RB1.4: Subproceso de Evaluación de Reglas de Enrollado (Rollup Rule Check Subproceso).** Evalúa un conjunto de reglas de enrollado con una regla de acción dada para una actividad. Los "Controles de Enrollado" para los hijos se evalúan en el "Subproceso de Evaluar Hijos para el Enrollado" para determinar qué hijos contribuyen a la evaluación. El "Subproceso de Evaluar Condiciones de Enrollado" se aplica a cada hijo contribuyente para determinar si su estado de seguimiento actual sigue el criterio de la regla de enrollado. El Subproceso de "Evaluación de Reglas de Enrollado" termina cuando evalúa una regla que indica que debe cambiarse el estado de la actividad.

El "Subproceso de Evaluación de Reglas de Enrollado" está especificado [por el siguiente pseudo-código](#). Describe sólo la lógica de evaluación de reglas, no se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.

1. **Subproceso de Evaluar Condiciones de Enrollado:** Aplica un conjunto de condiciones de enrollado a una actividad para determinar si se aplican. Cada condición se evalúa con la información de seguimiento actual de la actividad.

El "Subproceso de Evaluar Condiciones de Enrollado" está especificado [por el siguiente pseudo-código](#). Describe sólo la lógica de evaluación de condiciones, no se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.

2. **Subproceso de Evaluar Hijos para Enrollado:** Determina si la información de seguimiento para una actividad debería incluirse durante el enrollado. El subproceso comprueba los Controles de Enrollado (**ver SM.8**) de la actividad con una acción de enrollado deseada.

El "Subproceso de Evaluar Hijos para Enrollado" está especificado [por el siguiente pseudo-código](#). Describe sólo la lógica de evaluación, no se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.



- **RB1.5: Proceso de Enrollado Total (Overall Rollup Process):** Se usa para propagar los resultados de seguimiento a través de todo el árbol de actividad. Se aplica a:
 - Un árbol de actividad, delimitado por la actividad raíz del árbol.
 - La actividad en el árbol que finaliza y lanza el enrollado.

La raíz del árbol y la actividad actual especifican un camino de enrollado. El enrollado se aplica a todas las actividades del camino, empezando desde la madre de la actividad actual y siguiendo hacia atrás hasta la raíz.

Se aplican todos los procesos de enrollado en cada paso: "Medida de Enrollado", "Enrollado de Objetivo" y "Progreso de Actividad de Enrollado". No hay orden de aplicación de los procesos, excepto que primero se aplica la "Medida de Enrollado." Si el enrollado cambia la información de objetivo de una actividad que transfiere información de objetivo a un objetivo global compartido, este debe actualizarse antes de continuar el enrollado. Cuando empieza el proceso de enrollado, se lleva a cabo sobre todas las actividades del camino; el proceso no para si el enrollado no cambia ningún valor.

El "Proceso de Enrollado Total" está especificado [por el siguiente pseudo-código](#). Describe sólo la lógica de enrollado, no se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.

14.13.12. SR. Modelo de Comportamiento de Selección y Aleatoriedad (Selection and Randomization Behaviour Model)

La Secuenciación Simple identifica actividades para entregar al alumno entre las definidas en el árbol de actividades, que pueden estar definidas estáticamente o ser generadas dinámicamente. Los "Controles de Selección y Aleatoriedad" definen cómo se seleccionan y reordenan las hijas de una actividad antes y mientras son usadas en la secuenciación.

Una implementación debe invocar a los procesos de selección y aleatoriedad en los momentos adecuados para mostrar los siguientes comportamientos:

1. Los procesos de selección y aleatoriedad no tienen efectos si se aplican sobre una actividad que es hoja del árbol.
2. Los procesos de selección y aleatoriedad pueden dar como resultado una lista ordenada de hijas disponibles para una actividad identificada, remplazando al conjunto actual de Hijas Disponibles (**AM.1.1**) de la actividad.
3. El proceso de selección debe ser invocado antes del proceso de aleatoriedad, si ambos se aplican al mismo tiempo.



4. Las hijas en una actividad no deben ser seleccionadas o puestas en orden aleatorio si la actividad o cualquiera de sus descendientes están activas ("Actividad" vale "Activa": **AM.1.1**) o suspendidas ("Actividad" vale "Suspendida": **AM.1.1**).
5. La lista de "Hijas Disponibles" debe mantenerse inalterada durante toda la duración en que la actividad está activa o suspendida.
6. La lista de "Hijas Disponibles" debe mantenerse mientras que haya una posibilidad de otro intento sobre la actividad, a menos que se especifique con "Selection Timing" (**SM.9**) o Randomization Timing (**SM.10**) que puede aplicarse el proceso de selección o aleatoriedad, incluso si la actividad ya no está activa, sin reparar en si la actividad está suspendida o no.

Nota: Si el árbol de actividad es dinámico, los procesos de selección y aleatoriedad deben aplicarse a una actividad cada vez que se alteren sus descendientes.

7. En una travesía dirigida del árbol de actividad a través del "Subproceso de Flujo" (SB.3), cualquier actividad para la cual está especificado el "Selection Timing" o "Randomization Timing" debe invocar los procesos de selección y/o aleatoriedad antes de que las hijas de la actividad sean atravesadas. Para tal actividad, la travesía sólo se aplica a la lista de Hijas Disponibles en el orden definido por dicha lista.

Nota: Los procesos de selección y/o aleatoriedad deben tener lugar si la actividad está activada o no, pero antes de que sus hijas puedan ser atravesadas; el proceso no debe aplicarse a una actividad cuyas hijas no van a ser atravesadas.

8. **Una petición de Elección a través del Proceso de Petición de Elección (SB.9)**, puede tener como objetivo una actividad que es descendiente de una actividad para la cual se requiere selección o aleatoriedad, sólo si los procesos de selección y/o aleatoriedad se han aplicado porque se requerían para cada ancestro de la actividad objetivo.

Los procesos de selección y aleatoriedad están controlados por partes del modelo de definición de secuenciación ("Selection Controls" y "Randomization Controls").

Los procesos de selección y aleatoriedad no necesitan datos del modelo de seguimiento.

Los procesos de selección y aleatoriedad usan datos del modelo de estado de actividad ("Activity State Information").

Los procesos de selección y aleatoriedad pueden cambiar el atributo Hijas Disponibles (**AM.1.1**) de las actividades procesadas.



SR.1: Proceso de Seleccionar Hijos (Select Children Process). Se aplica a las hijas identificadas de una actividad. No devuelve ningún valor. Puede cambiar las "Hijas Disponibles" (**AM.1.2**) de la actividad identificada. Los atributos de definición de secuenciación Controles de Selección, controlan la selección de la actividad. El proceso se define como el "Proceso de Seleccionar Hijos": Del conjunto de todas las actividades hijas, selecciono unas, sin reemplazo, conservando el orden relativo.

El "Proceso de Seleccionar Hijos" está especificado [por el siguiente pseudo-código.](#)

No se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.

SR.2: Proceso de Hijas Aleatorias (Randomize Children Process). Se aplica a las hijas disponibles de la actividad. No devuelve ningún valor. Puede cambiar las "Hijas Disponibles" (**AM.1.2**) la actividad identificada.

Los atributos de definición de secuenciación Controles de Aleatoriedad, controlan el proceso de aleatoriedad. El proceso se define como Proceso de Hijas Aleatorias: Reordena aleatoriamente las hijas disponibles de la actividad.

El Proceso de Hijas Aleatorias está especificado [por el siguiente pseudo-código.](#) No se dice nada sobre cómo se implementa ni sobre cómo se trata la información. Únicamente describe el comportamiento esperado que exhibirá una implementación.

14.13.13. SB. Modelo de Comportamiento en Serie (Sequencing Behaviour Model)

El proceso de Secuenciación Simple (Simple Sequencing Process) es dirigido, en parte, por una "solicitud de secuenciación" para poder identificar el contenido de un elemento que debe ser presentado a un estudiante en un momento dado. El Simple Sequencing Process evalúa dicha petición para ver el modelo de contenido descrito en el árbol de actividades y con esos datos y viendo por dónde se encuentra el estudiante en ese momento (cual es su experiencia), pueda determinar que contenido le debe presentar. A este proceso de evaluar la solicitud e identificar el objeto de contenido adecuado (sino devuelve error) es a lo que se le llama "sequencing process" (proceso de secuenciación).

El Proceso de Secuenciación "Process sequencing" no presupone cómo ni cuando se generará una solicitud, tan sólo se debe asegurar de que la petición de secuenciación es válida y de que se envía o se presenta el objeto de contenido correspondiente cuando se debe. A parte de identificar dicho objeto de contenido, este proceso de secuenciación usa elementos del modelo de seguimiento (**TM**) y de la estructura del árbol de actividades global o total.

El "Proceso de Secuenciación Total o Global" (Overall Sequencing Process, **Ver OP**) es el que relaciona el proceso de secuenciación con los procesos de navegación, terminación, enrollado, selección y entrega.



El proceso de secuenciación se controla en parte, mediante el modelo de definición de la secuenciación, ver **SM**:

- Modos de Control de la Secuenciación
- Descripción de la Norma de Secuenciación.
- Descripción de Condiciones Límite.
- Descripción de Objetivos.

El proceso de secuenciación usa todas las partes del modelo de seguimiento, ver **TM**:

- TM.1.1: Información del progreso del objetivo
- TM.1.2: Información del progreso de la Actividad / Intento.

También usa datos del modelo del estado de la actividad, ver **AM**:

- AM.1: Modelo de información de Estado de Actividad.

El comportamiento del proceso de secuenciación está definido en términos de siete procesos, uno por cada posible solicitud de secuenciación, en términos también de un proceso de secuenciación total o global y de siete procesos más asociados para darle soporte:

- **Sequencing Rules Check Process:** aplica las reglas de secuenciación para determinar que acciones se deben tomar.
- **Check Activity Process:** aplica condiciones límite y deshabilita reglas para ver si se permite una determinada actividad.
- **Flow Tree Traversal Subprocess:** determina cual es la siguiente actividad del árbol, en preorden.
- **Flow Activity Sequencing Subprocess:** determina si se debe presentar o enviar una determinada actividad chequeando sus condiciones límites y sus reglas de secuenciación.
- **Flow Subprocess:** atraviesa el árbol de la actividad en una determinada dirección para encontrar la siguiente actividad.
- **Check Activity Process:** determina si una actividad individual está deshabilitada o viola las condiciones límites.
- **Choice Activity Sequence Subprocess:** determina si una petición de secuencia "Choice" se permite para una actividad.
- **Start Sequencing Request Process:** procesa una petición de secuencia "Start".



- **Resume All Sequencing Request Process:** procesa una solicitud "Resume All".
- **Continue Sequencing Request Process:** procesa una solicitud "Continue".
- **Previous Sequencing Request Process.**
- **Choice Sequencing Request Process.**
- **Retry Sequencing Request Process.**
- **Exit Sequencing Request Process,** procesan su solicitud correspondiente.

Todos estos procesos individuales forman parte del "Proceso de Petición de Secuenciación total o global" (overall Sequencing Request Process) que controla todo el comportamiento de la secuenciación.

- **SB.1 Petición de Secuenciación (Sequencing Requests):** El proceso de secuenciación responde a una de las siguientes solicitudes de secuenciación, cuyos nombres no tienen ningún significado ni semántica por sí mismos. Una solicitud de secuenciación lleva implícita la acción completa que se debe llevar a cabo cuando dicha solicitud sea procesada.
- **SB.2 Comportamiento de la Secuenciación (Sequencing Behaviour):** Aquí se describe como el procesador de la secuenciación interpreta una solicitud de secuenciación en combinación con los elementos del modelo de definición de secuenciación ("Sequencing Definition Model" **SM**), con los datos del modelo de seguimiento ("Tracking Model" **TM**), y el modelo del estado de la actividad ("Activity State Model" **AM**) para poder validar dicha solicitud e identificar así el objeto de contenido que debe ser enviado o presentado al estudiante, a no ser, claro está, que la solicitud sea de salida "Exit".

Cualquier implementación debe ser capaz de representar el proceso descrito arriba y exhibir así el comportamiento deseado.

- **SB.2.1 Subproceso del flujo transversal en el árbol (Flow Tree Traversal Subprocess):** La secuenciación se hace siguiendo un recorrido en preorden (transversal) en el árbol de actividades. Las diferentes peticiones de secuenciación atraviesan el árbol de la actividad hacia delante o hacia atrás. El "Flow Tree Traversal Subproceso" se usa para determinar cuál será la siguiente actividad en el recorrido, empezando en alguna actividad en concreto (o cual será actividad previa si vamos recorriendo el árbol hacia atrás). *Este proceso se define:* el proceso atraviesa el árbol bien hacia adelante o bien hacia atrás, a partir de una actividad especificada, para identificar la siguiente actividad (un

solo paso en el árbol). Si se va más allá del final del árbol o antes del principio del mismo se lanza una excepción. Sólo se aplican reglas de secuenciación que sean precondiciones relacionadas con el control del flujo del recorrido en el árbol.

El recorrido se controla mediante las "reglas de secuenciación" y los "Modos de Control". **(Ver SM.2)**

Por lo tanto, este subproceso se usa para identificar una actividad en la dirección indicada, y así poder devolver la siguiente actividad en el árbol. El "Flow Tree Traversal Subproceso" se especifica en [el siguiente pseudo-código](#)

- **SB.2.2 Subproceso del flujo transversal en la actividad (Flow Activity Traversal Subprocess):** La secuenciación atraviesa el árbol de actividades, aplicando las reglas de secuenciación y las condiciones límites. Este subproceso se usa para determinar si una actividad debe ser enviada, o si se debe examinar la siguiente actividad en el árbol. Este subproceso se basa o usa los siguientes apartados **SM.1, SM.2, SM.3, UP.4, UP.2 Y SB.2.1**.

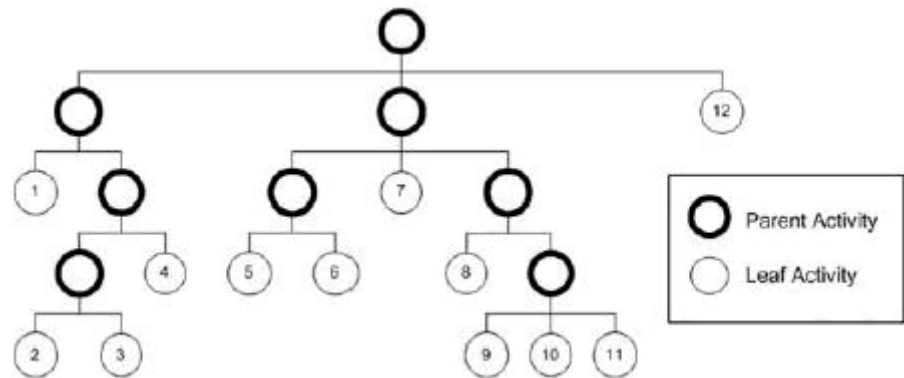
Definición: Este proceso atraviesa el árbol completo, usando un recorrido en preorden, partiendo de una actividad en particular. Si el "Sequencing Control Mode Flow" no está activo en el nodo padre de dicha actividad, se para sin identificar a la actividad que había que enviar. También se comprueba si no se exceden las Condiciones Límites impuestas a dicha actividad y se para si estos límites son excedidos. Se para también si la actividad está inhabilitada (SM.2: "Disable") o si está marcada como salida (SM.2: "Skipped"). Si la actividad no es una hoja del árbol, se atraviesa el árbol para encontrar la siguiente actividad.

Si el subproceso no tiene ningún problema al recorrer el árbol a partir de la actividad dada devolverá la siguiente actividad en el árbol sino devolverá una actividad que no se pueda enviar que indique donde falló el flujo de navegación en el árbol. Este subproceso se especifica en el [siguiente pseudo-código](#).

- **SB.2.3 Subproceso de flujo (Flow Subprocess):** Este subproceso atraviesa el árbol de actividades, desde una actividad designada de antemano, en una dirección específica y devuelve la actividad en la que se para si ésta está habilitada para ser enviada.

El proceso puede ser descrito como un recorrido (transversal) en el árbol. Se comienza en la Actividad Actual "Current Activity" (**AM.1.2**) y se usa el Subproceso de Flujo Trasversal en el Árbol "Flow Tree Traversal Subproceso" (**SB.2.1**) para ir a la siguiente actividad. Se sale o se termina si la actividad no puede ser

enviada, sino, se atraviesa el árbol de actividades usando el Subproceso del flujo de la Actividad "Flow Activity Sequencing Subproceso" (SB.2.2) para determinar si el recorrido debe terminar o porque la actividad deba o no ser enviada.



Si el ("Sequencing Control Flow" **SM.1**) tiene el valor cierto para todos los nodos padres de las actividades, y no hay otras definiciones de secuencia, se encontrarán las actividades hojas en el orden indicado en la figura anterior. Ver [el pseudo-código](#) que especifica este subproceso.

- **SB.2.4 Subproceso de Elección transversal de la Actividad (Choice Activity Traversal Subprocess):** Este subproceso se usa para determinar si una petición de secuenciación "Choice" está permitida en una actividad y se aplica a una sola actividad, comprobando las condiciones límite y las reglas de secuenciación. Devolverá "False" si se viola cualquier condición o regla o "True" si se puede alcanzar la actividad.

Definición: este proceso comprueba las condiciones límites para una actividad y para si se viola cualquier condición (**SM.3**), si la actividad está deshabilitada (**SM.2 "Disable"**) o si no se permite el recorrido en esa dirección (Ej.: "Stop Foward Traversal" **SM.2**). [Ver pseudo-código](#) que especifica dicho subproceso.

- **SB.2.5 Proceso de Petición Secuenciación "Start" (Start Sequencing Request Process):** Una solicitud de secuenciación "Start" se usa para identificar cual será la primera actividad que se entregará. Por eso este proceso se usa para recorrer el árbol de actividades e identificar cual será la primera actividad que se enviará al estudiante, aplicando, como no, las reglas de secuenciación y las condiciones límites.



El proceso de secuenciación usa el "Subproceso de Flujo" (**SB.2.3**), pero no usa, explícitamente, ninguna definición de secuenciación ni ningún dato del modelo de seguimiento.

Definición: El proceso recorre el árbol completo, en preorden (trasversal) hacia adelante, desde la raíz usando el "Flow Activity Sequencing Subproceso". Para cualquier actividad, si (**SM.1**) no está permitido para su padre, se para el recorrido sin identificar a la actividad que se debe enviar. También se comprueban las "Condiciones Límites" (**SM.3**) y se para el recorrido si cualquiera de ellas se violan o si la actividad está deshabilitada (**SM.2: "Disable"**) o está marcada como salida (**SM.2: "Skipped"**). Si la actividad no es una hoja, el árbol se recorrerá el árbol para encontrar la actividad siguiente. La primera actividad que sea hoja en el árbol será candidata a ser la actividad enviada. [Ver pseudo-código](#) del proceso.

- **SB.2.6 Proceso de Petición Secuenciación "Resume All" (Resume All Sequencing Request Process):** Una petición de secuenciación "Resume All" se usa para identificar la primera actividad a enviar intentando resumir la sesión de secuenciación anterior. Este proceso recorre el árbol de actividades, aplicando las reglas de secuenciación y las condiciones límites mediante el **UP.5**, para identificar la actividad que será enviada en primer lugar.

El proceso de secuenciación usa el atributo "Suspended Activity" (**AM.1.2**) para construir un camino de la raíz a la actividad donde la petición de navegación "Suspend All" fuera lanzada en la sesión de secuenciación anterior.

Definición: proceso recorre el árbol en preorden por completo, desde la raíz a través de un único camino, hacia la actividad designada como "Suspended Activity". Mientras recorre el camino, se ignoran los "Modos de Control de Secuenciamiento" (**SM.1**), pero si se comprueban las "Condiciones Límites" (**SM.3**) para cada actividad y se parará si supera algún límite o si en el recorrido se llega a alguna actividad que estuviera deshabilitada (**SM.2: "Disabled"**). Al final o se devuelve la actividad, o una indicación de que no se ha podido enviar ninguna actividad. [Ver el pseudo-código](#) del proceso.

- **SB.2.7 Proceso de Petición Secuenciación "Continue" (Continue Sequencing Request Process):** La petición de secuenciación "Continue" se usa para identificar la actividad a enviar, que será la siguiente a la actual en un recorrido del árbol hacia delante, aplicando las reglas de secuenciación y las condiciones límite.



Este proceso usa el subproceso de flujo (**SB.2.3**), pero no usa ninguna definición de secuenciación ni ningún dato del modelo de seguimiento.

Definición: El proceso recorre el árbol completo, en preorden (transversal) hacia adelante, desde la "Actividad Actual" (**AM.1.2**) y usando el "Subproceso de Flujo". Cada actividad que se encuentra es considerada un candidato para ser enviada. En cualquier actividad, si no está activo el "Sequencing Control Flow" (**SM.1**) para el padre, se para sin identificar a la actividad que será enviada. También se parará si se exceden las "Condiciones Límites" (**SM.3**), se llega a una actividad "Disabled" o "Skipped" (**SM.2**). Si la actividad no es una hoja, se recorre el árbol para encontrar la siguiente actividad. La primera encontrada será la candidata y o se devuelve esa actividad o una indicación de que no se devuelve ninguna. [Ver pseudo-código](#) del proceso.

- **SB.2.8 Proceso de Petición Secuenciación**

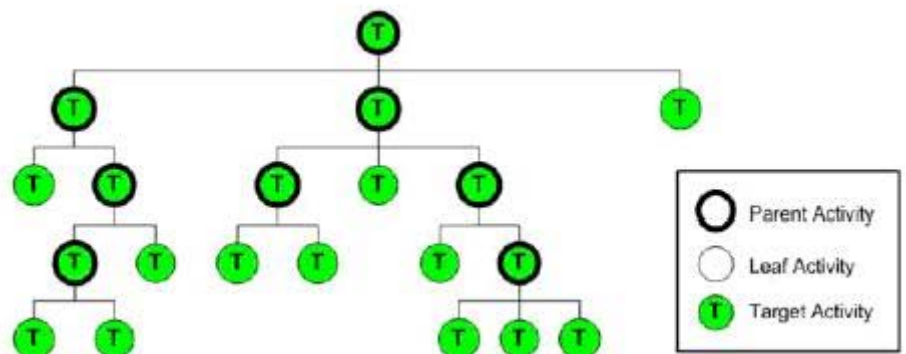
"Previous"(Previous Sequencing Request Process): La petición de secuenciación "Previous" se usa para identificar la actividad siguiente que deberá ser enviada comenzando en la actividad actual y recorriendo el árbol hacia atrás, aplicando, como siempre, las reglas de secuenciación y las condiciones límites.

Este proceso usa el subproceso de flujo (**SB.2.3**), pero no usa ninguna definición de secuenciación ni ningún dato del modelo de seguimiento.

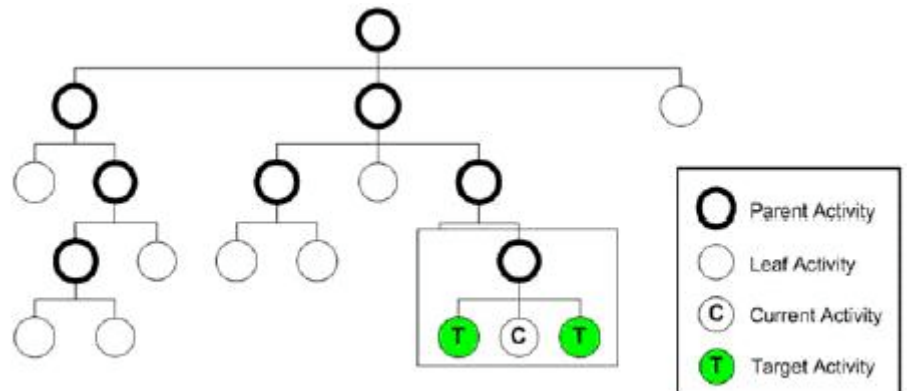
Definición: El proceso recorre el árbol completo, en preorden inverso trasversal, desde la Actividad Actual (**AM.1.2**) usando el Subproceso de Flujo. Cada actividad que se encuentra es considerada un candidato. En cualquier actividad, si no está activo el "Sequencing Control Flow" (**SM.1**) para la actividad padre, se para sin identificar a la actividad que será enviada. También se parará si se exceden las "Condiciones Límites" (**SM.3**), se llega a una actividad "Disabled" o "Skipped" (**SM.2**). Si se consideran los hijos de una actividad que tiene el atributo "Sequencing Control Forward Only" (**SM.1**), se recorren esas actividades hijas empezando desde el primer hijo en dirección hacia delante. Si la actividad no es una hoja, se recorre el árbol para encontrar la siguiente (es decir el anterior) actividad. La primera encontrada será la candidata y o se devuelve esa actividad o una indicación de que no se devuelve ninguna. [Ver pseudo-código](#) del proceso.



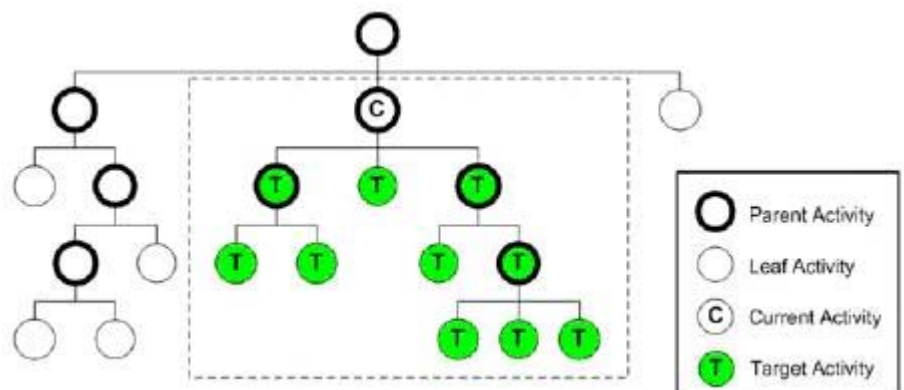
- La Actividad Actual y la actividad objetivo son la misma: Se comprueba si se puede entregar dicha actividad.



- Si la Actividad Actual y el objetivo son hermanas: Se recorre el árbol hacia el objetivo, ya sea hacia delante o hacia detrás.



- La actividad objetivo es un descendiente de la Actividad objetivo: se recorre hacia delante el árbol.





-

- [illegible]

- Que el atributo "Sequencing Control Flow" de la actividad sea "True", entonces dicha actividad llama al "Flow Subproceso" y se recorre el árbol hasta que se identifica a la actividad candidata para ser enviada o el proceso falla.



- Que dicho atributo sea igual a "False", entonces no habrá actividad candidata para ser enviada.

[Ver pseudo-código de este proceso.](#)

- **SB.2.10 Proceso de Petición de Secuenciación "Retry" (Retry Sequencing Request Process):** La solicitud de "Retry" se usa para restablecer de nuevo la actividad actual cuando hay un nuevo intento.

Este proceso usa un proceso que se explicará más adelante, el "Check Activity Process" (**UP.5**), pero no usa explícitamente definiciones de secuenciación o datos del modelo de seguimiento. *Definición:* este procedimiento comprueba el estado de la actividad actual mediante el "Check Activity Process" (**UP.5**), si dicha actividad está permitida y no es una hoja, se aplica el subproceso ya visto "Flow Subprocess" (**SB.2.3**) para determinar que actividad es la que se debe enviar. Si es una hoja será la actividad objetivo para ser enviada. Sino puede devolver la actividad devolverá una indicación que indique que la actividad no pudo ser enviada. [Ver pseudo-código](#)

- **SB.2.11 Proceso de Petición de Secuenciación (Sequencing Request Process):** Es el proceso de control general o total que maneja todas las peticiones de secuenciación y que sólo se encarga de enviar cada solicitud a un procesador adecuado. El procesador individual, por su parte, hará una de las siguientes cosas:

- Indica que el proceso no ha podido identificar a la actividad candidata para ser enviada: o porque no existe tal candidato (se esperará la siguiente solicitud de navegación) o porque existe pero se han excedido las condiciones límites.
- Indica que el proceso entero debe finalizar.
- Devuelve un candidato para que pueda ser enviado.
- Devuelve un error.

[Ver pseudo-código](#) de este proceso.



14.13.14. DB. Modelo de Comportamiento de Entrega (Delivery Behavior Model)

El proceso de secuenciación simple, que se viene describiendo a lo largo de estos capítulos, identifica una actividad cuyos recursos de contenido deben ser enviados al alumno (una petición de entrega). Pero antes de enviar el contenido, el proceso Secuenciación Simple debe validar que dichos recursos (obtenidos de la actividad identificada) puedan ser enviados, es decir, que se mantengan todas las condiciones necesarias para el envío y el intento por parte del estudiante por conseguirlas. Este proceso de validación e iniciación del envío del contenido es lo que se ha llamado proceso de envío "delivery process". Una vez que la petición de envío se valida, se envían los recursos de contenido correspondientes.

No se especifica el mecanismo usado para enviar estos recursos, ni el usado para actualizar los datos de seguimiento para los objetivos y las actividades del estudiante.

El OP relaciona el proceso de envío con los procesos de navegación, secuenciación, salida, selección... Además este proceso se controla por otros procesos o partes de estos ya vistos: "Tracking Model: Objective/Progress Information", "Activity State Model"...

Está formado por:

- DB.1 Comportamiento de Entrega (Delivery Behavior).
- DB.2 Proceso del Entorno del Envío de Contenido (Content Delivery Enviroment Process).

DB.1 Comportamiento de Entrega (Delivery Behavior).

Describe como un sistema de secuenciación interpreta una petición de envío, en combinación con los elementos del modelo de secuenciación y una instancia del modelo de seguimiento, para validar dicha petición de envío e iniciar el envío del contenido de la actividad.

Sólo son válidas para las solicitudes de entrega, las actividades que son hojas en el árbol.

El comportamiento de entrega se basa en el SM y en el TM.

- **DB.1.1 Proceso de Petición de Envío (Delivery Request Process):**
Se usa para validar la actividad identificada antes de enviarla. El proceso se aplica a: Un árbol de actividades (designado por su raíz) o una actividad hoja del árbol que haya sido identificada para el envío. Ambas, la raíz o la hoja, especifican un único camino a través del árbol que se llama camino de envío "delivery path". [Ver pseudo-código](#) del proceso.



DB.2 Proceso del Entorno del Envío de Contenido (Content Delivery Enviroment Process): Una vez que ha validado la petición de envío, se envía la actividad. El entorno del envío debe enviar los recursos, con el contenido de la actividad, al estudiante. El proceso de envío de este contenido no está especificado, por ejemplo, cómo recuperar dicho contenido de dónde está guardado, enviarlo como respuesta http...tampoco da seguridad de que el estudiante haya recibido lo que se envió.

Este proceso controla como comienza la actividad o como es accedida y como se guardan los datos para dicha actividad. Da por supuesto que la petición ha sido validada anteriormente y guarda la información necesaria para dar soporte a las peticiones de seguimiento y navegación. [Ver pseudo-código](#) para este proceso.

- **DB.2.1 Subproceso de la Actividad Suspendida limpia (Clear Suspended Activity Subproceso):** Este subproceso limpia el estado suspendido de una Actividad Suspendida y todos sus antecesores, para asegurar que el estado del árbol de actividades es actual. Para ello no usa ningún dato del modelo de definición de secuenciación ni del modelo de seguimiento. Lo que sí usa son partes del modelo del estado del modelo de la actividad: "Suspended Activity", que describe la actividad actual de la anterior sesión de secuenciación, la cual terminó con una petición de suspensión de toda la navegación. [Ver pseudo-código](#) de este subproceso.

14.13.15. UP. Procesos de Utilidad (Utility Processes)

Se usa para comprobar reglas y condiciones. Está formado por:

- UP.1 Proceso de Comprobación de las Condiciones Límites (Limit Conditions Check Process).
- UP.2 Proceso de Comprobación de las Reglas de Secuenciación (Sequencing Rules Check Process).
- UP.3 Proceso de Intentos de Terminación Descendientes (Terminate Descendent Attempts Process).
- UP.4 Proceso de Intento Final o "Fin" (End Attempt Process).
- UP.5 Proceso de chequeo de la actividad. (Check Activity Process).



- **UP.1 Proceso de Comprobación de las Condiciones Límites (Limit Conditions Check Process).**

Este proceso aplica las condiciones límites, del modelo de definición de secuenciación, para determinar si alguna de las condiciones límite de la actividad han sido violadas. Este Proceso se encuentra controlado por algunas partes del modelo de definición de secuenciación: "Limit Conditions", "Delivery Controls". Y usa además, algunas partes del modelo de seguimiento: "Activity/Attempt Progress Information", pero no usa el modelo del estado de la actividad.

Las condiciones límite se especifican mediante todos los atributos Condición Límite del modelo de definición de secuenciación (**SM.3**), que limitan el número intentos de acceso, la duración de dicho acceso o el límite de tiempo. Estas condiciones límite se evalúan usando los valores de los datos incluidos en los atributos del modelo de seguimiento: "Activity and Attempt Progress".

El orden en el que se evalúen las condiciones límites es arbitrario. [Ver pseudo-código](#) del proceso.

- **UP.2 Proceso de Comprobación de las Reglas de Secuenciación (Sequencing Rules Check Process).**

Este proceso evalúa un conjunto de reglas de secuenciación para una actividad, intentado identificar la acción que debe ser aplicada en cada caso.

Este proceso de comprobación está controlado por algunas partes del modelo de definición de secuenciación: "Sequencing Rules". Pero no usa el modelo de seguimiento ni el modelo del estado de la actividad.

El Proceso de Comprobación de las Reglas de Secuenciación utiliza el Subproceso de Comprobación de las reglas de Comprobación, que se verá a continuación, (**UP.2.1**) para evaluar cada regla. El proceso termina cuando hay una primera regla que evalúa a cierto, y devuelve la Acción de la Regla que evalúo a cierto "Rule Action" o Nil si ninguna evalúo a cierto. El orden de evaluación en el que se evalúa el conjunto de reglas de secuenciación, está definido en la "Descripción de las Reglas de Secuenciación de la Actividad" "Sequencing Rule Description" (**SM.3**). [Ver pseudo-código](#) del proceso.

- **UP.2.1 Subproceso de Comprobación de las Reglas de Secuenciación (Sequencing Rule Check Suprocess):** Evalúa un conjunto de condiciones en contraposición con la información de seguimiento actual que se tiene de la actividad. Este subproceso devolverá el valor cierto, si el conjunto de condiciones es conocido.



Este subproceso está controlado por algunas partes del modelo de definición de secuenciación: "Sequencing Rules", y también usa todas las partes del modelo de seguimiento: "Objective Information", "Activity/Attempt Progress Information". Pero no usa el modelo del estado de la actividad. [Ver pseudo-código](#) del subproceso.

- **UP.3 Proceso de Intentos de Terminación Descendientes (Terminate Descendent Attempts Process).**

Este proceso termina el intento (attempt) actual dentro de un conjunto de actividades para asegurarse de que el árbol de actividades esta actualizado. No devuelve ningún valor, sino que determina cuál es el antecesor común entre una actividad identificada y la actividad actual y luego aplica el Proceso **(UP.4)** "End Attempt Process" a cada actividad descendiente de ese antecesor común.

No usa datos del modelo de seguimiento ni del modelo de definición de secuenciación, pero sí usa algunas partes del modelo del estado de actividades: "Current Activity". [Ver-pseudo código](#) del proceso.

- **UP.4 Proceso de Intento Final o "Fin" (End Attempt Process).**

El resultado de la interacción del estudiante con una actividad se refleja en un conjunto de datos con los progresos de los objetivos y de los intentos. Este estándar no dice como ni cuando se establecen esos resultados, sin embargo, si supone que algunos objetos de contenido establecen directamente los datos del progreso de las actividades y los objetivos, aunque no todos los objetos deban tener esta habilidad de poner o establecer datos. Se usan los controles de envío: "Completion Set by Content and Objective Set by Content" para indicar si el objeto de contenido tiene o no esta capacidad de poner los datos.

El "nd Attempt Process" especifica los valores que se deben usar cuando los objetos de contenido no pueden establecer los valores por sí mismos. Los valores se establecen sólo cuando se sigue la actividad mediante el modelo de seguimiento **(T.M)**, que es el que tiene los valores por defecto por si no se establecen otros.

Los controles de entrega son los únicos encargados de establecer los valores en el modelo de seguimiento **(T.M)**. Cuando lo especifican los Controles de Entrega, los valores establecidos por el "End Attempt Process" colocan los valores de seguimiento. Sin embargo nunca se sobrescriben los valores establecidos por el objeto de contenido. [Ver pseudo-código](#) del proceso.



- **UP.5 Proceso de chequeo de la actividad. (Check Activity Process).**

El proceso se aplica a una determinada actividad ya identificada para ver si es una actividad permitida. Para ello se comprueban tanto las condiciones límites como las reglas de secuenciación "Disabled". Devuelve cierto si se viola cualquiera de esas condiciones o reglas. Este proceso aplica tanto el "Limit Conditions Check Process" (**UP.1**) como el "Sequencing Rules Check Process" (**UP.2**). [Ver pseudo código del proceso.](#)

14.13.16. Ver evaluación en Internet

14.14. BIBLIOGRAFIA

- 1.-IMS Global Learning Consortium : <http://www.imsproject.org>
- 2.-Edutec. Revista Electrónica de Tecnología Educativa. Núm. 13.



15. APÉNDICE D: PROCEDIMIENTOS A MODIFICAR SI SE ADOPTA EL ESTÁNDAR IMS LEARNING DESIGN

Administracion.java

-Crea un manifiesto vacío (imsmanifest.xml) con elementos vacíos: organizations y resources.

RegistroServlet.java

-Solo crea el curso y llama a: curso.titulo()

ElegirOrganizacionServlet.java

-Permite elegir la organización por lo que cambiará o se eliminará porque las organizaciones desaparecen del manifiesto.

indiceServlet.java

-Busca si hay mas de una organización para poner el botón "cambiar organización" en el índice.
->Si no se puede cambiar de organización, se eliminaría el if.

VolcadoAlumnoServlet.java

-Llama al método: curso.referencia(id), donde id es el identificador del ítem. La modificación iría en función del cambio en la clase curso.

IncluirCursoServlet.java

-Presenta la lista de los cursos que forman parte de la universidad.xml. Llama al método: curso.titulo().

ListaCursosProvServlet.java

-Crea una lista de los cursos provisionales: recorre la lista de cursos provisionales buscando su nombre en el elemento <identifier>. Luego saca el título con: curso.titulo().

registroProfesorServlet.java

-Crea una lista de los cursos que imparte un profesor: recorre la lista de cursos buscando su nombre en el elemento <identifier>. Luego saca el título con: curso.titulo().

CrearExamenServlet.java

-Crea un examen y lo añade al curso: curso.nuevorecurso(examen,href), marca la organización actual con: curso.organizacion(idOrg), saca el elemento ítem que será el examen: curso.item(idItem), le asocia el recurso (examen) que acaba de crear y actualiza los cambios: curso.guardar().

editorOrganizacionServlet.java

-Edita la organización seleccionada. Habría que modificarlo/sustituirlo para editar elementos "learning design".

**editarRecursoServlet.java**

-Edita un recurso. Llama al curso para obtener el recurso: `curso.recurso(idRec)`; también para crear un nuevo recurso: `curso.nuevorecurso(tipo,nombre)`; y por último lo mete en la organización: `curso.organizacion(idOrg)`, saca el ítem: `curso.item(idItem)`, le asocia el recurso y actualiza los cambios: `curso.guardar()`.

diasSinConexionServlet.java

-Obtiene el título del curso: `curso.titulo()`, para presentarlo por pantalla al mostrar la lista de alumnos que no han accedido al curso dado en un tiempo específico.

CursoServlet.java

-Marca como actual el ultimo item visitado por el alumno (en la sesión anterior) para mostrárselo por pantalla. Saca el identificador de la organización del ultimo item visitado: `curso.organizaciondeitem(idPagina)`, si no existe, obtiene las organizaciones (`curso.organizaciones()`) para mostrarlas al alumno y que elija una (si existen varias).

editorCursoServlet.java

-Edita varias partes del curso: `curso.titulo()`, `curso.organizaciones()`, `curso.tituloorganizacion(idOrg)`, `curso.pontitulo(titulo)`, `curso.borrarorganizacion(borrarOrg)`. Actualizarlo/ampliarlo para que permita editar las diferentes partes del learning design.

indiceServlet.java

-Accede a las organizaciones (`curso.organizaciones()`) para comprobar si tiene que poner el botón "cambiar organizaciones" en el marco del índice.

cursoSeguimientoServlet.java

-Presenta enlaces a los alumnos matriculados en un curso (mostrando su título: `curso.titulo()`), y presenta los exámenes para editarlos: `curso.examenes()`.

PaginaAuxServlet.java

-Muestra por pantalla un ítem, por eso usa varios métodos de la clase curso: `curso.organizacion(idOrg)`, `curso.idItem()`, `curso.apunta(id)`, `curso.anterior()`, `curso.siguiente()`, `curso.referencia()`, `curso.prerrequisitos(id)`, `curso.tipo()`, `curso.capitulo()`.

editarItemServlet.java

-Muestra el formulario para editar un ítem. Usa varios métodos de la clase curso.

cursoSeguimientoAux.jsp

-Establece el curso como atributo de la sesión: `sesion.setAttribute("curso", curso)`.



16. APÉNDICE E: BIBLIOGRAFÍA

- **Professional XML** - Mark Birbeck, Michael Kay, Stev Livingstone, Stephen F. Mohr, Jonathan Pinnock, Brian Loesgen, Steven Livingston, Didier Martin, Nikola Ozu, Mark Seabourne, David Baliles Editorial Wrox Press Inc
- **Professional Xml (Programmer to Programmer): 2nd Edition** - Mark Birbeck (Editor), Nikola Ozu, Jon Duckett, Jon Duckett, Stephen Mohr, Kevin Williams, Oli Gauti Gudmundsson, Daniel Marcus, Pete Kobak, Evan Lenz, Zoran Zaev, Steve Livingstone, Jonathan Pinnock, Keith Visco, Wrox Author tea, Andrew H. Watt, Andrew Watt, Stephen Mohr, Kevin Williams, Oli Gauti Gudmundsson, Raja Mani, Daniel Marcus, Peter Kobak, Evan Lenz, Mark Birbeck, Brian Hickey, Zoran Zaev, Steven Livingstone, Jonathan Pinnock, Keith Visco Editorial Wrox Press Inc
- **Professional Java Servlets 2.3** – Andrew Harbourne-Thomas, Sam Dalton, Simon Brown, Bjarki Holm, Tony Loton, Meeraj Kunnumpurath, Subrahmanyam Allamaraju, John Bell, Sing Li Editorial Wrox Press Inc
- <http://java.sun.com/>
- <http://www.w3.org/TR/xpath>
- <http://www.w3.org/XML/>
- <http://www.w3.org/MarkUp/>
- <http://www.w3.org/DOM/>
- <http://www.programacion.com/tutorial/javaxml/2/>
- <http://www.imsglobal.org>



17. APÉNDICE F: AUTORIZACIÓN

Doña Beatriz Vera Mateos,
Doña Esther Valverde Escorial,
y Don Jorge Juan Llorente Marcos,
autorizan a la Universidad Complutense a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Beatriz Vera Mateos

Esther Valverde Escorial

Jorge Juan Llorente Marcos



18. APÉNDICE G: LISTA DE PALABRAS CLAVE

- IMS
- Curso virtual
- Aula virtual
- Universidad virtual
- E-aula
- E-learning
- Content Packaging
- Meta-datos
- Flash
- Estandares